ALGORITHMS FOR APPROXIMATION II

Based on the proceedings of the Second International Conference on Algorithms for Approximation, held at Royal Military College of Science, Shrivenham, July 1988.

Edited by

J.C. MASON
Professor of Computational Mathematics,
Royal Military College of Science, Shrivenham

and

M.G. COX
Senior Principal Scientific Officer,
National Physical Laboratory, Teddington

These proceedings were supported by a contract from the **US Air Force (European Office of Aerospace Research and Development):** Reference CSP-88-1026.

CHAPMAN AND HALL LONDON 1989

DEDICATION

We dedicate this book to the memory of Professor Jerry L. Fields of University of Alberta, who died recently. Jerry made significant contributions to rational approximation and special functions, and was a very generous friend.

PREFACE

This book tells the story of four summer days in Shrivenham in 1988, when 88 people came to the Second International Conference on Algorithms for Approximation at the Royal Military College of Science (RMCS) from July 12th to 15th. Of course the book only tells part of the story. It cannot easily convey the good humour and very happy atmosphere that prevailed, and it does not catalogue the many friendships that were made, the excellent meals and refreshments that were enjoyed, and the plethora of social activities that took place.

The conference was organised for two main reasons. Firstly, there were many requests for a Second Shrivenham Conference from those who attended the First in 1985. Secondly, it had clearly been established at the First Conference that there was a demand and need for a regular meeting which emphasised the algorithmic aspects and applications of approximation. There was also a continuing need for more conferences in the UK in the general area of approximation.

The meeting itself was run under the auspices of Cranfield Institute of Technology, of which RMCS is a faculty, and financial support was gratefully received from the US Office of Aerospace Research and Development. The organising committee was Maurice Cox (NPL) and Anne Daman and John Mason (RMCS), and the RMCS local organisers were very ably assisted by Elizabeth Smith and Pamela Moore. It was a loss to RMCS and to the conference organisation when Anne Daman left early in 1988 to seek her fortune in the USA, but there were celebrations when she returned in July as Mrs Anne Trefethen to receive her PhD degree and attend the conference with her husband.

The ten invited speakers, chosen by the organising committee, covered a broad spectrum of topics and came from a wide range of countries. All of their papers appear in these proceedings. The UK speakers were Dr John Gregory (Brunel), Professor John Mason (RMCS), Professor Michael Powell (Cambridge) and Professor Alastair Watson (Dundee). Also from Europe came Professor Wolfgang Dahmen (Berlin) and Professor Tom Lyche (Oslo). The speakers from the USA were Dr Eric Grosse (Bell Labs), Professor Larry Schumaker (then Texas A and M, now Vanderbilt) and Professor Lloyd Trefethen (MIT). Finally from Canada came the opening speaker, Professor Ian Barrodale (Barrodale Associates, Victoria), who set the tone of all the invited talks with a paper of excellent content and very entertaining delivery. It is also appropriate at this point to mention Dr James Lyness (Argonne), who as guest speaker at the conference dinner, added further academic distinction and good humour to the occasion.

The remainder of the conference programme was made up of submitted papers, of which forty were selected for two parallel sessions. Within this book you will find the thirty-three of these papers which were finally offered and accepted for publication after a refereeing procedure. There was general agreement on the high standard of the conference talks, and we believe that this is reflected in the overall quality of the published papers.

It will be noted that the last paper in the book, by Eric Grosse, is a substantial catalogue of algorithms for approximation, and we believe that this will prove to be a very useful and popular reference. Moreover we should not be suprised to see this becoming the catalyst for a series of such offerings. Unlike the proceedings of the first conference, the present volume does not include a "software panel". However, this is more than compensated for by the inclusion of the above-mentioned catalogue and the separate publication (by Chapman and Hall) of a volume of proceedings of a Symposium on "Scientific Software Systems", held at RMCS on the day before the conference (July 11th). The latter volume of seventeen papers covers the broad area of numerical software, as well as a wide range of software and hardware requirements. It includes, amongst many contributions which might interest current readers, two papers with immediate application to approximation, viz "Linear algebra support modules for approximation and other software" by M G Cox and "An advisory expert system for curve and surface fitting" by Anne E Trefethen.

The forty-one papers in the present volume have been arranged into three primary sections: I Development of Algorithms, II Applications, and III Catalogue of Algorithms. The first two sections have been subdivided into eight groups: 1. Spline approximation, 2. Polynomial and piecewise polynomial approximation, 3. Interpolation, 4. Smoothing and constraint methods, 5. Complex approximation, 6. Computer-aided design and geometric modelling, 7. Applications in numerical analysis, and 8. Applications in other disciplines.

Such a division into sections, while giving the book a useful structure, is somewhat arbitrary, and we apologise to any authors who may feel that their work has been incorrectly categorised. Several papers could have been placed in up to three groups (especially spline approximation, piecewise polynomial approximation, and computer-aided design). Moreover the CAD group, which we have placed in the Applications section could perfectly well have been placed in Section I. Although there is no group headed "nonlinear approximation", there are several nonlinear algorithms (in Section II in particular), and of course the complex algorithms (in group 5 and elsewhere) could have come under this heading.

We must conclude with some essential but broad expressions of our gratitude. Firstly, we thank the multitude of staff at Royal Military College of Science, National Physical Laboratory, US Air Force European Office of Aerospace Research, and Chapman and Hall (publishers) who contributed in so many different ways to make the conference a great success and to print the abstracts and proceedings. Secondly, we thank all the speakers and authors of invited and contributed papers, without whose industry and patience this volume would not have existed.

As soon as conference number II was over, we were tempted to start thinking in terms of a conference series. We certainly intend that there should be further conferences on "Algorithms for Approximation", since the first two generated so much good research and goodwill. All that we need are energy, time and resources and, of course, your support.

John Mason
Shrivenham

Maurice Cox
Teddington

May 1989

CONTRIBUTORS

R.W.ALLEN
Allen Clarke Research Centre, Plessey Research Ltd, Caswell, Towcester, Northants, NN12 8EQ.

G.T.ANTHONY
Division of Information Technology and Computing, National Physical Laboratory, Teddington, Middlesex, TW11 OLW.

E.ARGE
Institut fur Informatikk, University of Oslo, P.O.BOX 1080, Blindern, 0316 Oslo 3, Norway.

I.BARRODALE
Barrodale Computing Services Ltd, 1677 Poplar Avenue, Suite 200, Victoria B.C., V8P 4K5, Canada.

T.B.BOFFEY
Department of Statistics and Computational Mathematics, University of Liverpool, Liverpool, L69 3BX.

K.W.BOSWORTH
Department of Mathematics and Statistics, Utah State University, Logan, Utah, 84322-3900, U.S.A.

M.BOZZINI
Dipartimento di Matematica, Università di Lecce, Lecce, Italy.

L.BRUTMAN
Department of Mathematics and Computer Science, University of Haifa, Haifa 31999, Israel.

M.D.BUHMANN
Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge, CB3 9EW.

M.G.COX
Division of Information Technology and Computing, National Physical Laboratory, Teddington, Middlesex, TW11 OLW.

M.DAEHLEN
Institut fur Informatikk, University of Oslo, P.O.BOX 1080, Blindern, 0316 Oslo 3, Norway.

W.DAHMEN
Fachbereich Mathematik (WE3), Freie Universität Berlin, Arnimallee 2-6, D-1000 Berlin 33, West Germany.

L.M.DELVES
Department of Statistics and Computational Mathematics, University of Liverpool, Liverpool, L69 3BX.

N.DYN
School of Mathematical Science, Tel Aviv University, Tel Aviv, Israel.

T.A.FOLEY
Computer Science Department, Arizona State University, Tempe, Arizona 85287, U.S.A.

A.B.FORBES
Division of Information Technology and Computing, National Physical Laboratory, Teddington, Middlesex, TW11 OLW.

W,FREEDEN
Institut für Reine und Angewandte Mathematik, R.W.T.H. Aachen, Templergraben 55, D-5100 Aachen, W.Germany.

F.N.FRITSCH
Computing and Mathematics Research Division, Lawrence Livermore National Laboratory, University of California, P.O.BOX 808, Livermore, California 94550, U.S.A.

M.FRONTINI
Dipartimento di Matematica, Politecnio di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy.

R.H.J.GMELIG MEYLING
Department of Applied Mathematics, University of Twente, P.O.BOX 217, 7500 AE Enschede, The Netherlands.

M.von GOLITSCHEK
Institut für Angewandte Mathematik, Universität Würzburg, West Germany.

J.A.GREGORY
Department of Mathematics and Statistics, Brunel University, Uxbridge, Middlesex, UB8 3PH.

E.GROSSE
A.T. & T., Bell Laboratories, Room 2T-504, 600 Mountain Avenue, Murray Hill, New Jersey 07974, U.S.A.

P.M.HARRIS
Division of Information Technology and Computing, National Physical Laboratory, Teddington, Middlesex, TW11 OLW.

M.HEILMANN
Lehrstuhl Mathematik VIII, Universität Dortmund, Postfach 500500, D-46 Dortmund 50, West Germany.

A.ISERLES
Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge, CB3 9EW.

H.M.JONES
Division of Information Technology and Computing, National Physical Laboratory, Teddington, Middlesex, TW11 OLW.

P.E.KOCH
Division of Mathematical Sciences, Norwegian Institute of Technology, Trondheim, Norway.

M.A.LACHANCE
Department of Mathematics and Statistics, University of Michigan-Dearborn, Dearborn, Michigan 48128-1491, U.S.A.

L.LENARDUZZI
Istituto Applicazioni Matematica ed Informatica, C.N.R., Via Ampere 56, 20131 Milano, Italy.

D.LEVIN
School of Mathematical Sciences, Tel Aviv University, 69978 Tel Aviv, Israel.

T.LYCHE
Institut fur Informatikk, University of Oslo, P.O.BOX 1080, Blindern, 0316 Oslo 3, Norway.

J.C.MASON
Applied and Computational Mathematics Group, Royal Military College of Science, Shrivenham, Swindon, Wiltshire, SN6 8LA.

J.G.METCALFE
Allen Clarke Research Centre, Plessey Research ltd, Caswell, Towcester, Northants, NN12 8EQ.

K.MØRKEN
Institut fur Informatikk, University of Oslo, P.O.BOX 1080, Blindern, 0316 Oslo 3, Norway.

M.W.MÜLLER
Lehrstuhl Mathematik VIII, Universität Dortmund, Postfach 500500, D-46 Dortmund 50, West Germany.

S.P.NØRSETT
Division of Mathematical Sciences, Norwegian Institute of Technology, Trondheim, Norway.

G.D.OLAOFE
Mathematics Department, University of Ibadan, Ibadan, Oyo State, Nigeria.

G.OPFER
Institut für Angwandte Mathematik, Universität Hamburg,
Bundesstrasse 55, D-2000 Hamburg 13, W.Germany.

C.POTIER
Department Informatique, E.N.S.T., 46 Rue Barrault, 75634 Paris-Cedex 13, France.

M.J.D.POWELL
Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge, CB3 9EW.

C.J.PURSGLOVE
Department of Statistics and Computational Mathematics, University of Liverpool, Liverpool, L69 3BX.

E.QUAK
Department of Mathematics, Vanderbilt University, Nashville, Tennessee 37235, U.S.A.

C.RABUT
INSA, Service de Mathematiques, 31077 Toulouse Cedex, France.

C.RADEMACHER
Institut für Angewandte Mathematik, Universität Bonn, Wegelerstrasse 6 D-5300 Bonn 1, West Germany.

S.RIPPA
School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv, 69978 Tel-Aviv, Israel.

G.RODRIGUEZ
Dipartimento di Matematica, Università di Cagliari, Cagliari, Italy.

A.RON
Department of Computer Science, University of Wisconsin, Madison, Wisconsin, U.S.A.

J.M.SANZ-SERNA
Departamento de Matemàtica Aplicada y Computaciòn Facultad de Ciencias, Universidad de Valladolid, Valladolid, Spain.

K.SCHERER
Institut für Angewandte Mathematik, Universität Bonn, Wegelerstrasse 6, D-5300 Bonn 1, West Germany.

L.L.SCHUMAKER
Department of Mathematics, Vanderbilt University, Nashville, Tennessee 37235, U.S.A.

S.SEATZU
Dipartimento di Matematica, Università di Cagliari, Cagliari, Italy.

P.W.SMITH
IMSL, 2500 City West Boulevard, Houston, Texas 77042, U.S.A.

P.T.P.TANG
Division of Mathematics and Computer Science, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439, U.S.A.

F.DE TISI
Dipartimento di Matematica, Università di Milano, Milan, Italy.

L.N.TREFETHEN
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusettes 02139, U.S.A.

ANNE E. TREFETHEN
Thinking Machine Corporation, Cambridge, Massachusetts, U.S.A.

C.VERCKEN
Department Informatique, E.N.S.T., 46 Rue Barrault, 75634 Paris-Cedex 13, France.

G.WALZ
Fakultät für Mathematik und Informatik, Universität Mannheim, D-6800 Mannheim 1, West Germany.

G.A.WATSON
Department of Mathematical Sciences, University of Dundee, Dundee, Scotland, DD1 4HN.

S.J.WILDE
Applied and Computational Mathematics Group, Royal Military College of Science, Shrivenham, Swindon, Wiltshire, SN6 8LA.

J.WILLIAMS
Department of Mathematics, University of Manchester, Manchester, M13 9PL.

C.ZALA
Barrodale Computing Services Ltd, 1677 Poplar Avenue, Suite 200, Victoria, British Columbia, V8P 4K5, Canada.

CONTENTS

I. DEVELOPMENT OF ALGORITHMS

1. Spline Approximation

Constrained spline approximation of functions and data based on constrained knot removal by E.Arge, M.Daehlen, T.Lyche*, and K.Morken.

Near real-time spline fitting of long sequences of uniformly spaced data, by G.T.Anthony[+] and M.G.Cox.

An algorithm for knot location in bivariate least squares spline approximation by M.Bozzini[+] and F.de Tisi.

A knot placement stategy for least squares spline fitting based on the use of local polynomial approximations by M.G.Cox, P.M.Harris[+] and H.M.Jones.

An algorithm for nonlinear splines with non-negativity constraints by G.Opfer[+].

Spline curve fitting of digitized contours by C.Potier[+] and C.Vercken.

A B-spline approximation algorithm for quasi-interpolation or filtering by C.Rabut[+].

On knots and nodes for spline interpolation by P.W.Smith[+].

2. Polynomial and Piecewise Polynomial Approximation

A basis for certain spaces of multivariate polynomials and exponentials by W.Dahmen*.

Monotone piecewise cubic data fitting by F.N.Fritsch[+].

Direct and converse results on simultaneous approximation by the method of Bernstein-Durrmeyer operators by M.Heilmann and M.W.Müller[+].

Orthogonality and approximation in a Sobolev space by A.Iserles[+], P.E.Koch, S.P.NØrsett and J.M.Sanz-Serna.

Piecewise polynomial approximation of polynomial curves by M.A.Lachance[+].

Calculation of the energy of a piecewise polynomial surface by E.Quak[+] and L.L.Schumaker.

3. Interpolation

Radial basis function interpolation on an infinite regular grid by M.D.Buhmann+ and M.J.D.Powell*.

The Fourier operator of even order and its application to an extremum problem in interpolation by L.Brutman+.

On multivariate polynomial interpolation by N.Dyn+ and A.Ron.

Algorithms for the constuction of data dependent triangulations by N.Dyn, D.Levin and S.Rippa+.

Algorithms for computing best parametric cubic interpolation by C.Rademacher and K.Scherer+.

4. Smoothing and Constraint Methods

Data fitting by penalized least squares by M.Von Golitschek and L.L.Schumaker*.

A semi-infinite programming algorithm for constrained best approximation by K.W.Bosworth+.

Inference region for a method of local approximation by using the residuals by M.Bozzini and L.Lenarduzzi+.

5. Complex Approximation

Numerical methods for Chebyshev approximation of complex-valued functions by G.A.Watson*.

A fast algorithm for linear complex Chebyshev approximation by P.T.P.Tang+.

II. APPLICATIONS

6. Computer Aided Design and Geometric Modelling

Uniform subdivision algorithms for curves and surfaces by N.Dyn, J.A.Gregory* and D.Levin.

Approximation by spheres by T.B.Boffey+, M.G.Cox, L.M.Delves and C.J.Pursglove.

Interpolation of scattered data on a spherical domain by T.A.Foley+.

Least squares best fit geometric elements by A.B.Forbes+.

Uniform piecewise approximation on the sphere by W.Freeden+ and J.C.Mason.

7. Applications in Numerical Analysis

Approximation theory and numerical linear algebra by L.N.Trefethen*.

An algorithm for computing minimum norm solutions of the finite moment problem by M.Frontini+, G.Rodriguez and S.Seatzu.

Numerical solution of the biharmonic equation using different types of bivariate spline functions by R.H.J.Gmelig Meyling+.

Quadrature solution of integral equations: a uniform treatment of Fredholm and Volterra equations by G.O.Olaofe+.

Increasing the convergence modulus of an asymptotic expansion: an algorithm for numerical differentiation by G.Walz+.

Approximation and parameter estimation in ordinary differential equations by J.Williams+.

8. Applications in Other Disciplines

Applications of discrete l_1 methods in science and engineering by C.Zala and I.Barrodale*.

Constrained complex approximation algorithms in communication engineering by J.C.Mason*, A.E.Trefethen and S.J.Wilde.

Integration of absolute amplitude from a decibel B-spline fit by R.W.Allen+ and J.G.Metcalfe.

A nonlinear least squares data fitting problem arising in microwave measurement by M.G.Cox and H.M.Jones+.

A complex minimax algorithm for phase-only adaptation in antenna arrays by J.C.Mason and S.J.Wilde+.

III. CATALOGUE OF ALGORITHMS

A catalog of algorithms for approximation by E.Grosse*.

* Invited speaker.   + Speaker.

I

DEVELOPMENT OF ALGORITHMS

1. Spline Approximation

CONSTRAINED SPLINE APPROXIMATION OF FUNCTIONS AND DATA BASED ON CONSTRAINED KNOT REMOVAL

ERLEND ARGE, MORTEN DÆHLEN, TOM LYCHE, KNUT MØRKEN
Institute of Informatics
University of Oslo

Abstract Two of the authors (Lyche and Mørken) have recently developed a knot removal strategy for splines in B-spline format, which has been applied successfully to the approximation of functions and data. In this paper we show how general constraints can be incorporated in this strategy and also in approximation methods based on knot removal. In our implementation of the knot removal strategy, two fundamental properties of B-splines were central—knot insertion or subdivision and the fact that the usual L^p-norms for splines can be approximated well by some simple discrete norms. Together with the fact that a B-spline expansion is a convex combination of the coefficients, these properties are also the key features in our treatment of constraints.

Key words: Splines, B-splines, Constrained approximation, Knot insertion, Knot removal, Discrete norms, Quadratic optimization.

1. Introduction

In Lyche and Mørken (1988), a strategy for removing knots from a B-spline function without perturbing the spline more than a prescribed tolerance was developed. It was also shown how this strategy could be used successfully to compute approximations to large sets of discrete data, by applying knot removal to some simple initial approximation like a cubic spline representation of the piecewise linear interpolant to the data. This also provides a method for approximating functions as long as they can be sampled sufficiently often. The knot removal technique was extended to parametric curves and surfaces in Lyche and Mørken (1987b).

Our general approach to constrained spline approximation will follow the same pattern as for unconstrained approximation. First we compute an initial approximation satisfying the constraints, but in general requiring a large number of parameters for its representation. Then we remove knots from this spline, but all the time making sure that we do not violate the constraints or the prescribed tolerance.

The literature on constrained approximation has grown considerably in the last few years, see Fontanella (1987) and Utreras (1987) for two recent surveys, and also the bibliography compiled by Franke and Schumaker (1987). We will consider a general class of constraints which requires one or more derivatives or the integral of the approximation to be bounded by general functions.

In the remaining part of this section we introduce our notation and give some fundamental results concerning splines. In Section 2 we discuss how various constraints can be written in terms of linear inequality and equality constraints involving the B-spline coefficients on a suffiently large knot vector. In order to carry out constrained knot removal, it is desirable to have an initial approximation which satisfies the constraints. The initial approximation scheme should also cope with the situation where the data do not satisfy the constraints. We discuss these questions in Section 3. Knot removal is the topic of Section 4. We review the unconstrained knot removal strategy in Lyche and Mørken (1988) and explain how it can be extended to handle constraints. The paper ends with a section discussing examples of constrained approximation using the proposed technique.

1.1 Notation

All splines in this paper are represented in terms of B-splines. We denote the i'th B-spline of order k on the knot vector $\boldsymbol{t}$ by $B_{i,k,\boldsymbol{t}}$ and assume the B-splines to be normalized to sum to one. If $\boldsymbol{t}$ contains $m+k$ elements with none occurring more than k times, then we can form m linearly independent B-splines on $\boldsymbol{t}$. These B-splines span a linear space of splines defined by

$$\mathbf{S}_{k,\boldsymbol{t}} = \Big\{ \sum_{i=1}^{m} d_i B_{i,k,\boldsymbol{t}} \;\Big|\; d_i \in \mathbf{R} \text{ for } i = 1, 2, \ldots, m \Big\}.$$

In this paper we will assume that $m \geq k$, and that $t_k < t_{k+1}$ and $t_m < t_{m+1}$, and we will only be interested in the spline functions on the interval $[t_k, t_{m+1}]$.

1.2 Knot insertion

The central ideas of this paper are consequences of the fact that the B-spline coefficients can model the spline they represent with arbitrary precision. This is based on so called subdivision or knot insertion techniques, so let us consider the basis for this. A spline on a knot vector $\boldsymbol{\tau}$ can also be represented as a spline on any knot vector $\boldsymbol{t}$ ıhat contains all the knots of $\boldsymbol{\tau}$. The reason for this is that if $\boldsymbol{\tau}$ is a subsequence of $\boldsymbol{t}$, then $\mathbf{S}_{k,\boldsymbol{\tau}}$ is a subspace of $\mathbf{S}_{k,\boldsymbol{t}}$. Let g be a spline in $\mathbf{S}_{k,\boldsymbol{\tau}}$ with coefficient vector c relative to $\boldsymbol{\tau}$ and coefficient vector b relative to $\boldsymbol{t}$. Then c and b are related by the equation $b = Ac$, where A is the knot insertion matrix of order k from τ to t. This matrix has properties very similar to the B-spline collocation matrix, see Jia (1983), Lyche and Mørken (1987a, 1988), and Lyche (1988). One consequence of these properties is that as more and more knots are inserted, the B-spline coefficients converge to the given spline g, see Cohen and Schumaker (1985) and Dahmen (1986), where it is shown that the convergence is quadratic in the knot spacing. This means that if we work on a sufficiently fine knot vector, we do not lose much by using the coefficients as a representation of the spline.

1.3 Discrete norms

The ability of the B-spline coefficients to mimic the spline they represent is also exemplified by the fact that certain simple combinations of the B-spline coefficients provide good approximations to the L^p-norms of the spline. Specifically, if $f = \sum_{i=1}^m d_i B_{i,k,t}$ is a spline on a knot vector of length $m+k$ as above, we can define a family of discrete norms called the (ℓ^p, t)-norms on $\mathbb{S}_{k,t}$ by

$$\|f\|_{\ell^p,t} = \begin{cases} \left\{\sum_{i=1}^m |d_i|^p (t_{i+k} - t_i)/k\right\}^{1/p}, & \text{for } 1 \le p < \infty; \\ \max_i |d_i|, & \text{for } p = \infty. \end{cases}$$

Equivalently, if we define the diagonal $m \times m$ matrix $E_t^{1/p}$ with the i'th diagonal element given by

$$(E_t^{1/p})_{i,i} = \begin{cases} \left\{(t_{i+k} - t_i)/k\right\}^{1/p}, & \text{for } 1 \le p < \infty; \\ 1, & \text{for } p = \infty; \end{cases}$$

then we also have

$$\|f\|_{\ell^p,t} = \|E_t^{1/p} d\|_p.$$

Here $\|\cdot\|_p$ denotes either the usual L^p norm of a function or the ℓ^p norm of a vector. The significance of the (ℓ^p, t)-norms is due to the fundamental inequalities

$$D_k^{-1}\|f\|_{\ell^p,\tau} \le \|f\|_p \le \|f\|_{\ell^p,t} \le \|f\|_{\ell^p,\tau},$$

where $\tau \subseteq t$ and $f \in \mathbb{S}_{k,\tau}$. The leftmost inequality is due to de Boor (1976a and 1976b). The number D_k depends only on k, and numerical experiments have shown that $D_k \sim 2^k$. As an example we have $D_4 \approx 10$.

The (ℓ^p, t)-norm is of course also a norm on any subspace of $\mathbb{S}_{k,t}$. If a spline in a subspace of $\mathbb{S}_{k,t}$ is given, then to compute its (ℓ^p, t)-norm it must be represented as a spline in $\mathbb{S}_{k,t}$ by knot insertion or degree elevation.

We will be using the (ℓ^∞, t)-norm to estimate the relative importance of the interior knots during knot removal, and the error of an approximation will be measured in this norm since it gives an upper bound on the L^∞-norm. The method we will employ to compute spline approximations will be best approximation in the (ℓ^2, t)-norm. This norm converges to the L^2-norm for continuous functions as the knot spacing goes to zero, see Lyche and Mørken (1988), and it can even be shown that the best approximation in the $(\ell^2, \boldsymbol{t})$-norm converges to the best approximation in the L^2-norm. The main advantage of working with this discrete norm instead of the L^2-norm is computational efficiency.

Compared with other $(\ell^p, \boldsymbol{t})$-norms, the $(\ell^2, \boldsymbol{t})$-norm has the advantage that best approximation leads to a simple linear system of equations. When it comes to constrained approximation this may not be so crucial, and approximation in the $(\ell^1, \boldsymbol{t})$-norm and the $(\ell^\infty, \boldsymbol{t})$-norm are interesting alternatives. To develop efficient algorithms for constrained spline approximation in these norms will be the subject of future work by the authors.

2. Constraints

Our approach to constrained approximation is also based on the fact that the B-spline coefficients model the spline they represent. Let $f = \sum_{i=1}^{m} d_i B_{i,k,t}$ be a spline in $\mathbb{S}_{k,t}$ to be approximated by a spline $g = \sum_{i=1}^{n} c_i B_{i,k,\tau}$ in a subspace $\mathbb{S}_{k,\tau}$, and denote by b the coefficient vector of g relative to $\boldsymbol{t}$, so that $b = Ac$. We will consider constraints of the form

$$\begin{aligned} E_1 b &\geq v_1, \\ E_2 b &= v_2, \end{aligned} \tag{1}$$

where E_1 and E_2 are rectangular matrices and v_1 and v_2 are vectors. (The notation $u \geq w$ for vectors u and w denotes the component-wise inequalities $u_i \geq w_i$ for $i = 1,\ 2,\ \ldots$.) In other words, the constraints can be expressed directly as restrictions on linear combinations of the coefficients relative to the $\boldsymbol{t}$ knot vector. The constraints characterize the set of permissible spline functions,

$$\mathsf{F}_{k,t} = \Big\{ \psi = \sum_{i=1}^{m} b_i B_{i,k,t} \;\Big|\; E_1 b \geq v_1 \;\&\; E_2 b = v_2 \Big\}.$$

As mentioned above, we will use best approximation in the $(\ell^2, \boldsymbol{t})$-norm to compute spline approximations. The typical optimization problem to be solved is therefore

$$\min_{\psi \in \mathsf{F}_{k,t} \cap \mathbb{S}_{k,\tau}} \|f - \psi\|_{\ell^2, t}, \tag{2}$$

and a matrix formulation of this problem is given in Section 4.2. This problem is a quadratic minimization problem with linear constraints. Such problems are studied extensively in the optimization literature, see e.g. Fletcher (1987). We emphasize that even though the approximation g is a spline on the knot vector $\boldsymbol{\tau}$, we minimize a weighted ℓ^2-norm of the B-spline coefficients of the error on the $\boldsymbol{t}$ knot vector which for good results should contain 'many' knots.

In practice, any constraints that can be handled by the quadratic minimization routine that is used to solve (2) are admissible, but in this paper we restrict our attention to linear constraints of the type (1). It should be noted here that constrained problems for parametric curves and surfaces often lead to nonlinear constraints, cf. Ferguson, Frank and Jones (1988).

It has been observed that bounds on a spline g or its derivatives can be replaced by a finite number of linear inequality constraints on the B-spline coefficients of g, see e.g. Cox (1987). In general though, the replacing constraints will be stronger than the original ones so that the set of feasible solutions is overly restricted. Recall that during knot removal we also have a requirement that the error should be less than the tolerance. This additional constraint together with the constraint on the B–spline coefficients may result in a problem having no feasible solutions. To weaken the constraints, we propose to use knot insertion and constrain the B-spline coefficients on a knot vector which has many knots compared to the number of oscillations in the spline to be approximated. In this way the discrepancy between the original and replacing constraints is reduced.

As an example, consider the polynomial $T(x) = 4x^3 - 3x + 1$ on the interval $[-1, 1]$ as a given function to be approximated on the Bezier knot vector $\boldsymbol{\tau}$ with four knots at -1 and 1, with L^∞-error less than ϵ and with nonnegativity as the constraint. The function T is the cubic Chebyshev polynomial with the constant 1 added to make it nonnegative. If we choose the approximation g to be T, then we solve the problem exactly. However, a general purpose algorithm for solving this type of problem would probably not discover that T is nonnegative and a feasible solution. Our approach implements the ideas above. A sufficient condition for nonnegativity is that the B-spline coefficients on some knot vector are nonnegative, and instead of requiring the L^∞-error to be less than ϵ we require the $(\ell^\infty, \boldsymbol{t})$-error to be less than ϵ for a suitable $\boldsymbol{t}$. Note that on the Bezier knot vector $\boldsymbol{\tau}$, one of the B-spline coefficients equals -4 so this knot vector is not suitable as $\boldsymbol{t}$, since the nonnegativity constraint would lead to a $(\ell^\infty, \boldsymbol{t})$-error of at least 4. If we insert the $2q - 1$ interior knots $(-1 + 1/q, -1 + 2/q, \ldots, -1/q, 0, 1/q, 2/q, \ldots, 1 - 1/q)$ into $\boldsymbol{\tau}$, where q is a positive integer, then it is easily seen that on this refined knot vector $\boldsymbol{t}^q$, all the B-spline coefficients of T are greater than $-2/q^2$. Therefore, for the set of feasible solutions

$$\mathsf{F}_{4,\boldsymbol{t}^q} = \Big\{ \psi = \sum_i b_i B_{i,4,\boldsymbol{t}^q} \;\Big|\; \|T - \psi\|_{\ell^\infty, \boldsymbol{t}^q} \le \epsilon \ \& \ c_i \ge 0 \text{ for all } i \Big\}$$

to be nonempty, we must have $2/q^2 \le \epsilon$ or $q \ge \sqrt{2}/\sqrt{\epsilon}$.

In the rest of this section we assume that we have a knot vector $\boldsymbol{t}$ on which the original constraints have been replaced by constraints on the B-spline coefficients in a satisfactory way. We now discuss some specific constraints, and we start by considering nonnegativity constraints in general. As was indicated in the example, the obvious way to implement straightforward nonnegativity is to require the B-spline coefficients on $\boldsymbol{t}$, of an approximation $g = \sum_{i=1}^m b_i B_{i,k,\boldsymbol{t}}$, to be nonnegative,

$$b_i \ge 0, \qquad \text{for } i = 1,\ 2,\ \ldots,\ m.$$

In many cases one only wants nonnegativity locally, say on an interval $[a, b]$ which is contained in $[t_k, t_{m+1}]$. This is accomplished by requiring nonnegativity of only the coefficients that multiply a B-spline with part of its support in $[a, b]$. To be able to strictly enforce such conditions, it may be necessary to refine $\boldsymbol{t}$ by including a and b as knots of multiplicity $k-1$.

A simple generalization of the nonnegativity constraint is to require

$$b_i \geq e, \qquad \text{for } i = 1, 2, \ldots, m,$$

for some real number e. The next step is then to require $g \geq h$ for a spline $h = \sum_{i=1}^{m_1} \hat{e}_i B_{i,k,t_1}$ on a knot vector $\boldsymbol{t}_1$. To be able to compare h, f and an approximation g we then replace $\boldsymbol{t}$ by $\boldsymbol{t} \cup \boldsymbol{t}_1$ and assume that on the new $\boldsymbol{t}$ knot vector the constraining spline is given by $h = \sum_{i=1}^{m} e_i B_{i,k,t}$. Then we would simply require

$$b_i \geq e_i, \qquad \text{for } i = 1, 2, \ldots, m.$$

These more general constraints are easily restricted to some interval just as above.

All of these constraints have restricted the approximation from below. If a restriction from above is required, this is most conveniently expressed as $-b_i \geq -e_i$.

There are many other types of constraints that can be written as linear equality and inequality constraints.

- Monotonicity and convexity.
  A monotone approximation g is characterized by $g' \leq 0$ or $g' \geq 0$ and a convex approximation by $g'' \geq 0$. By the well known differentiation formula for B-splines, these constraints can be implemented similarly to the nonnegativity constraints and also generalized in the same way. In general, any derivative of order less than k can be constrained in this manner.
- Interpolation.
  The approximation can be required to interpolate a given value y at a point x leading to an equality constraint of the form,

  $$\sum_{i=1}^{m} b_i B_{i,k,t}(x) = y.$$

  It is well known that at most k B-splines are nonzero at x so that this sum contains at most k terms. Any derivative can of course be interpolated in a similar way.
- Integral constraints.
  The integral of the approximation can be constrained by requiring

  $$\int g = \sum_{i=1}^{m} b_i(t_{i+k} - t_i)/k \geq e.$$

The equality follows because the integral of $B_{i,k,t}$ is known to be $(t_{i+k} - t_i)/k$. If only the integral over a small interval $[a, b]$ is to be constrained, one could insert $k-1$-tuple knots at a and b.

- Smoothness.
  An approximation g with a required smoothness is obtained by restricting the multiplicity of the knot vector on which g is defined.
- Discontinuities.
  Jumps in derivatives can be enforced by using some of the above techniques. If we want a sharp edge at the point a for instance, we must first ensure that a occurs with multiplicity $k-1$ in the knot vector. We can then for example enforce $g'(x) \geq 1$ for $x < a$ and $g'(x) \leq 0$ for $x > a$. In fact, it is easily seen that it is sufficient to restrict only three coefficients.

Any linear combination of the constraints above is also a valid constraint. As an example one could force the approximation to take on the value of its integral at a given point by combining interpolation and integration constraints.

It should be noted that the requirement that the (ℓ^∞, t)-error should be less than the tolerance can be implemented as a constraint of the above type. However, it seems more efficient to implement this constraint in the same manner as in unconstrained knot removal, cf. Section 4.

3. The initial approximation

In general, constrained approximation of data by constrained knot removal, will require two steps as described in the introduction. First, an initial spline approximation to the data satisfying the constraints is computed, and then constrained knot removal is applied to this initial approximation. The purpose of the second step is to remove those knots in the initial approximation that are redundant relative to a given tolerance. This step will be described in Section 4. Here we will discuss the problem of computing an initial approximation.

In general we also determine an initial approximation in two stages. First, the data are converted to a spline ϕ which does not necessarily satisfy the constraints. Then, on the knot vector t of ϕ, we compute a spline approximation f to ϕ satisfying the constraints.

Let us discuss each stage in more detail. For the first stage we are given some data to be approximated. These data can be in different forms. It may be a set of discrete data, a general function or a spline function. The purpose of Stage 1 is to convert the data into a spline ϕ of order k on a knot vector t. If the data is a set of discrete points (x_i, y_i) with the abscissae increasing, we can let ϕ be either the linear interpolant to the data, the Schoenberg variation diminishing spline, or we can determine ϕ by using unconstrained knot removal or some other suitable method. Similarly, if the data is a general function or a spline of order greater than k, we can approximate it by a spline of order k by using the same method. In either case we end up with a spline function ϕ of order k on a knot vector $\boldsymbol{t}$ of length $m+k$ with B-spline coefficients $(a_i)_{i=1}^{m}$.

In constrained approximation there will in general be a conflict between satisfying the constraints and making the error smaller than the tolerance, since the original data may not satisfy the constraints. As an example consider the case where the original data take on the value -1 at a point, but the constraints require the approximation to be nonnegative everywhere and the tolerance is 0.01. To cope with situations of this kind, we need Stage 2 to adjust the spline ϕ above to a spline f on the same knot vector that satisfies the constraints. This is done by simply minimizing the $\ell^2, \boldsymbol{t}$-norm of the error and enforcing the constraints. In other words, the spline f solves the quadratic optimization problem

$$\min_{\psi \in \mathsf{F}_{k,t}} \|\phi - \psi\|_{\ell^2, t} \tag{3}$$

which is just a special case of (2). This is equivalent to the problem

$$\min_{z \in \mathbf{R}^m} \|E_t^{1/2}(z-a)\|_2$$

subject to

$$E_1 z \geq v_1,$$
$$E_2 z = v_2.$$

The attitude taken here is that it is more important to satisfy the constraints than to keep the error smaller than the tolerance. This spline will then be given as input to the knot removal process.

It should be noted that this step is not always necessary. Consider once again a discrete set of nonnegative data. If we take the piecewise linear interpolant to the data as ϕ, then this spline will automatically satisfy the constraints and we can set $f = \phi$. The same applies with global monotonicity and convexity constraints if we approximate the discrete data by piecewise linear interpolation or by the Schoenberg variation diminishing spline.

In some situations the minimization problem (2) will succeed in finding a g in $\mathsf{F}_{k,t}$ even if $f \notin \mathsf{F}_{k,t}$. This would for example be the case when $\phi \geq -10^{-6}$, the tolerance is 0.01 and the constraint is nonnegativity.

4. Knot removal

With the initial approximation f, the tolerance, and the constraints given, we can start to remove knots. More precisely we have the following problem.

Constrained knot removal problem. Given a polynomial order k, a knot vector $\boldsymbol{t}$, a set of linear constraints characterizing a subset $\mathsf{F}_{k,t}$ of $\mathbf{S}_{k,t}$, a spline f in $\mathbf{S}_{k,t}$ and a tolerance ϵ; find a knot vector $\boldsymbol{\tau} \subseteq \boldsymbol{t}$ which is as short as possible and a spline $g \in \mathbf{S}_{k,\tau} \cap \mathsf{F}_{k,t}$ such that $\|f - g\|_{\ell^\infty, t} \leq \epsilon$.

To discuss this problem we first review unconstrained knot removal.

4.1 Review of unconstrained knot removal

In knot removal one is given a spline f in $\mathbf{S}_{k,t}$ and a tolerance ϵ; the goal is to determine a spline g in a subspace $\mathbf{S}_{k,\tau}$ of $\mathbf{S}_{k,t}$ of lowest possible dimension, such that $\|f-g\| \leq \epsilon$. The norm used here is in principle arbitrary, but in Lyche and Mørken (1988) we used the $(\ell^\infty, \boldsymbol{t})$-norm.

Since finding the shortest possible $\boldsymbol{\tau}$ seems very difficult, the method only attempts to find an approximate solution. The idea is to compute a ranking of the interior knots of $\boldsymbol{t}$ according to their significance in the representation of f, and then try to remove as many knots as possible according to this ranking. More specifically, for each interior knot t_i its weight w_i is computed as the absolute error in best $(\ell^\infty, \boldsymbol{t})$ approximation to f from the space $\mathbf{S}_{k,t\setminus\{t_i\}}$. Clearly, if t_i is redundant then $w_i = 0$. (This description is correct if all the interior knots are simple; multiple interior knots cause some complications.) We could then rank the knots according to their weights w_i, but this would not work very well when many knots have more or less equal weights. To compensate for this, knots with similar weights are grouped together. The first group consists of the knots with weights between 0 and $\epsilon/2$, the second group those with weights between $\epsilon/2$ and ϵ and so on. The knots in each group are listed in the order in which they occur in the knot vector. Suppose that there are 40 knots in the first group and 30 knots in the second group, and that we want to remove 50 knots. Those 50 knots would then be the 40 knots in the first group plus 10 knots from the second group. The 10 knots from the second group would be every third knot in the order that they occur in the group.

It is not possible to determine from the weights themselves how many knots can be removed without the error exceeding the tolerance. Therefore, the exact number of knots to be removed is determined by a binary search. First, half the knots are removed and an approximation is computed as the best approximation in the $(\ell^2, \boldsymbol{t})$-norm, together with the error in the $(\ell^\infty, \boldsymbol{t})$-norm. If the error is too large, the approximation is discarded and we try to remove 1/4 of the knots instead. If the error is acceptable, we save the approximation and try to remove 3/4 of the knots. This process is continued until the exact number of knots that can be removed has been determined.

By running through this process, a spline approximation g with knot vector $\boldsymbol{\tau}$ is determined. However, it turns out that it is usually possible to remove even more knots. The knots $\boldsymbol{\tau}$ of g are of course also knots of f and now we can compute a ranking of the knots of $\boldsymbol{\tau}$ as their significance in the representation of g. This ranking can then be used to remove more knots from $\boldsymbol{t}$ to obtain an approximation to f with even fewer knots than $\boldsymbol{\tau}$. The process terminates when no more knots can be removed without the error exceeding the tolerance.

The knot removal process outlined above constitutes a convenient basis for general spline approximation methods. Since we have a method for removing knots from a spline, we just have to produce a good approximation to a function or to discrete data without worrying too much about the number of knots since this can be reduced afterwards. Possible initial approximation schemes include piecewise linear approximation, cubic Hermite interpolation, Schoenberg variation diminishing approximation and many more.

4.2 Constrained knot removal

The general philosophy used in unconstrained knot removal can also be applied to the constrained case. The following changes in the strategy used in the last subsection are necessary for solving the problem stated at the beginning of this section.

- The fixed knot approximation method used during knot removal has to preserve the constraints.
- The ranking procedure also includes spline approximation, and in general the constraints must be taken into account when the weights are computed.

Constrained ranking is currently under investigation by the authors, and we will not discuss it any further in this paper. In the examples in Section 5 we will use the same ranking procedure as in the unconstrained case.

The fixed knot approximation method was introduced in Section 2. Suppose that we have decided to remove some knots and have a knot vector $\boldsymbol{\tau}$ which is a subsequence of $\boldsymbol{t}$. We want to find an approximation g on $\boldsymbol{\tau}$ to f which satisfies the constraints. We determine g by the optimization problem (2). If $\boldsymbol{\tau}$ has length $n+k$, then the B-spline coefficients of g must solve the quadratic minimization problem

$$\min_{z\in\mathbf{R}^n} \|E_t^{1/2}(Az-d)\|_2$$

subject to

$$\begin{aligned} E_1Az &\geq v_1, \\ E_2Az &= v_2. \end{aligned}$$

The knot removal process will continue until it is not possible to remove any more knots without either violating the tolerance or the constraints.

5. Examples

In this section we give three examples of the use of the constrained approximation scheme outlined in the preceding sections. The quadratic optimization problems involved have been solved by general purpose library software. Since most of the constraints cause the matrices E_1 and E_2 to be very sparse, performance can be improved considerably both in terms of storage and CPU-time by tailoring the optimization routines to this type of problems. This we consider to be an important area of future research.

5.1 One-sided approximation

The method outlined in the previous sections can be used to construct one-sided approximations to functions. In this example a tolerance $\epsilon = 10^{-2}$ is given, and we want to find a cubic spline approximation g to the function

$$\Phi(x) = \max(\sin(\pi x), 0), \qquad \text{for } x \in [0, 4],$$

subject to the constraint $g(x) - \Phi(x) \geq 0$ for all $x \in [0, 4]$. To construct an initial cubic spline approximation to Φ, let

$$f(x) = \sum_{i=1}^{209} (\Phi(t_i^*) + 10^{-3}) B_{i,4,t}(x),$$

where

$$t_i^* = \frac{1}{3}(t_{i+1} + t_{i+2} + t_{i+3}), \qquad i = 1,\ 2,\ \ldots,\ 209,$$

and the knot vector t consists of the numbers $(0.02 * j)$ for $j = 0,\ 1,\ \ldots,\ 200$, with knots of multiplicity four at 0 and 4, and knots of multiplicity three at 1, 2 and 3.

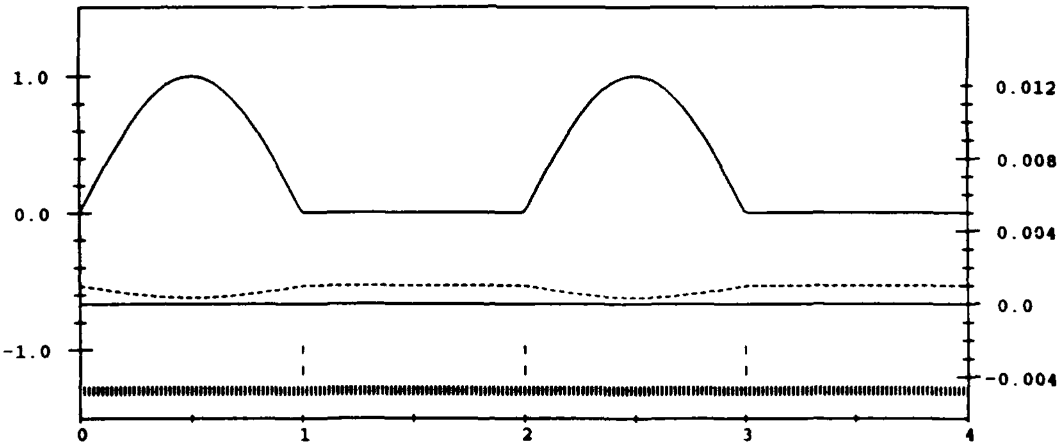


Figure 1. Initial one-sided approximation to a given function.

The spline f is the Schoenberg variation diminishing approximation to $\Phi + 10^{-3}$ which satisfies the constraints, but has far too many knots. Figure 1 shows the approximation f (solid curve) and the error $f - \Phi$ (dashed curve). The location of the knots of f are indicated at the bottom of Figure 1.

With the constrained knot removal procedure described in the previous sections, an approximation g to f is found subject to the constraint $g - f \geq 0$ and such that $g - f \leq \epsilon - 10^{-3}$. That is, the approximation g to Φ is within the given tolerance $\epsilon = 10^{-2}$ and $g - \Phi$ is positive. Figure 2 shows g (solid curve)

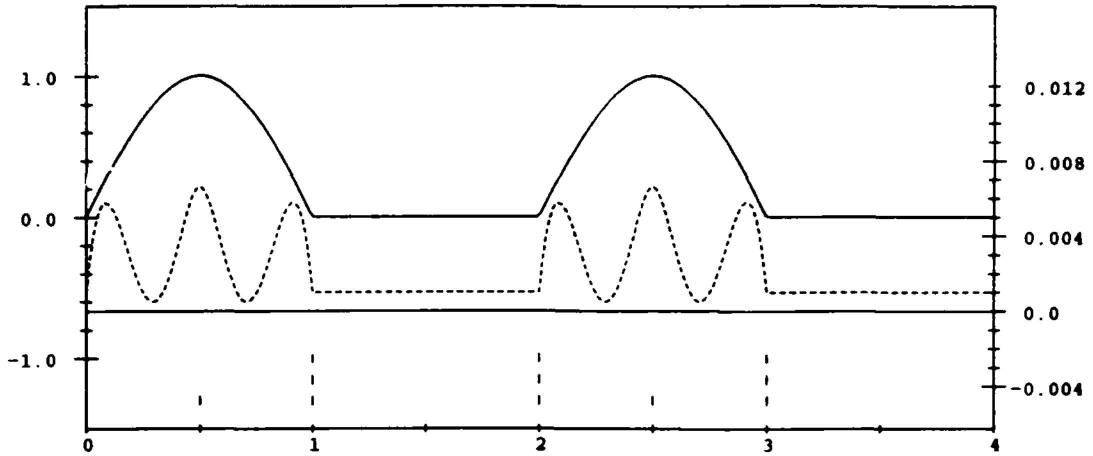


Figure 2. Final one-sided approximation after knot removal.

and the error $g - \Phi$ (dashed curve) together with the location of the interior knots left by the constrained knot removal method. The initial knot vector has been reduced to

$$\tau = (0,0,0,0,0.5,1,1,1,2,2,2,2.5,3,3,3,4,4,4,4).$$

5.2 Convex approximation.

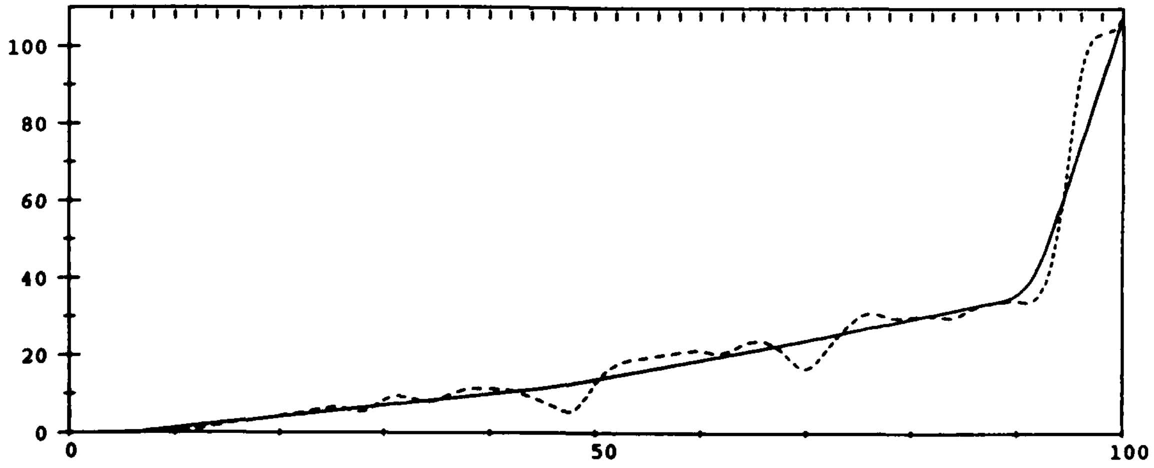


Figure 3. The initial convex approximation to the turbine data.

The dashed curve shown in Figure 3 is a cubic spline approximation to a set of data points

$$(x_i, y_i) = (2i, y_i), \qquad \text{for } i = 1,\ 2, \ldots,\ 50,$$

where y_i is measured electricity when $2i$ percent of maximum power is forced upon a particular water turbine. A large number of turbines have been tested and data measured, and all turbines have similar curves depending on the turbine type, the size of the turbine and various other factors. These curves make it possible to choose the right turbines given the water resources and the need for electric power. Physical considerations indicate that such curves should be convex, and with a tolerance of $\epsilon = 1.0$ there is clearly scope for knot removal, so it is natural to apply knot removal with convex constraints.

The dashed curve ϕ shown in Figure 3 is in fact the Schoenberg variation diminishing approximation to the data on the knot vector

$$\boldsymbol{t} = (t_i)_1^{54} = (2,2,2,2,6,8,\ldots,94,96,100,100,100,100).$$

The interior knots are indicated at the top of Figure 3. Our problem is to construct a spline approximation g to ϕ subject to the constraints

$$g(t_1) = \phi(t_1), \qquad g(t_{54}) = \phi(t_{54}), \qquad g''(x) \geq 0 \quad \text{for all } x \in [2,100].$$

First, we computed an initial spline approximation f on $\boldsymbol{t}$ which solves (3) with

$$\mathsf{F}_{4,\boldsymbol{t}} = \Big\{ \sum_j d_j B_{j,4,\boldsymbol{t}} \;\Big|\; d_1 = a_1 \;\&\; d_{50} = a_{50} \;\&\; \Delta d_j \geq 0, \text{ for } 3 \leq j \leq 50 \Big\},$$

where (a_i) and (d_i) are the B-spline coefficients of ϕ and f on $\boldsymbol{t}$, and Δd_j is defined by

$$\Delta d_j = \sigma_{j-1} d_j - (\sigma_{j-1} + \sigma_j) d_{j-1} + \sigma_j d_{j-2},$$

where $\sigma_i = (t_{i+3} - t_i)/3$. The spline f is shown in Figure 3 (solid curve).

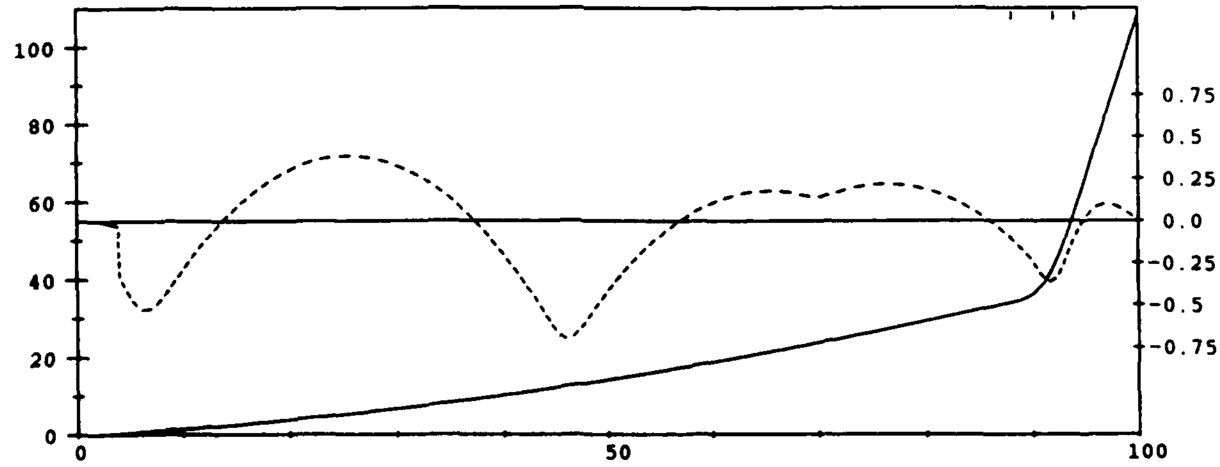


Figure 4. The final convex approximation after knot removal.

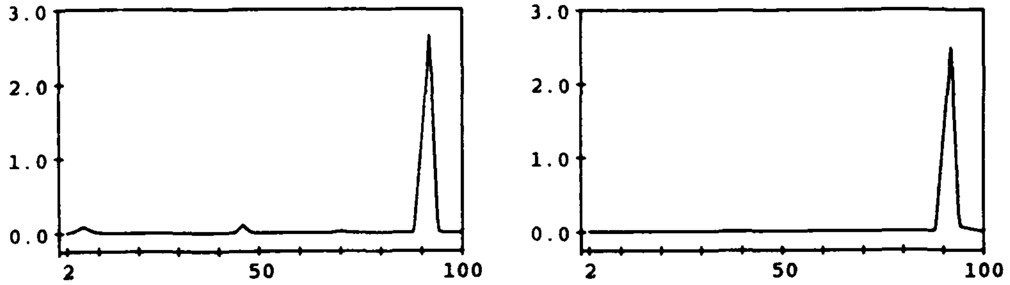


Figure 5. The second derivative of the initial (left) and final convex approximations.

Applying constrained knot removal to f with the same constraints and with tolerance $\epsilon = 1.0$, we get the final cubic spline approximation g. This spline is shown in Figure 4 (solid curve) together with the error function $g - f$ (dashed curve). The three remaining interior knots are indicated at the top of the figure.

Figure 5 shows the second derivative of f and g, and we observe that g is smoother than f. We also observe that the computed convex approximation to the given data consists almost of two straight lines.

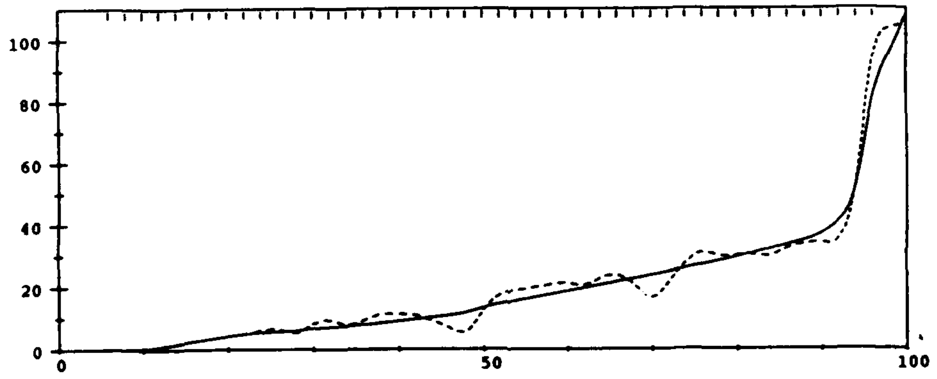


Figure 6. Initial approximation to the turbine data with the first derivative bounded from below by a positive spline.

5.3 Curved constraints

This example illustrates that it is possible to introduce curved constraints into the knot removal method. The initial data are the values of the cubic spline ϕ on the knot vector $\boldsymbol{t} = (t_i)_1^{54}$ of the previous example. We are also given a quadratic spline h on the same knot vector (the dashed curve in Figure 8). Our problem is to construct a spline approximation g to ϕ with as few knots as possible, subject to the constraints

$$g(t_1) = \phi(t_1), \qquad g(t_{54}) = \phi(t_{54}), \qquad g'(x) \geq h(x) \quad \text{for all } x \in [2, 100].$$

The tolerance is 1.0 as in the previous example.

First, we compute an initial spline approximation f on $\boldsymbol{t}$ which solves (3) with

$$\mathsf{F}_{4,t} = \Big\{ \sum_j d_j B_{j,4,t} \;\Big|\; d_1 = a_1 \;\&\; d_{50} = a_{50} \;\&\; (d_j - d_{j-1})/\sigma_j \geq e_j \text{ for } 2 \leq j \leq 50 \Big\},$$

where a_j and (e_j) are the B–spline coefficients of ϕ and h on $\boldsymbol{t}$, and $\sigma_j = (t_{j+3} - t_j)/3$. The result is shown in Figure 6, with ϕ dashed and f as a solid curve. We note that f is monotone since we have chosen a positive h.

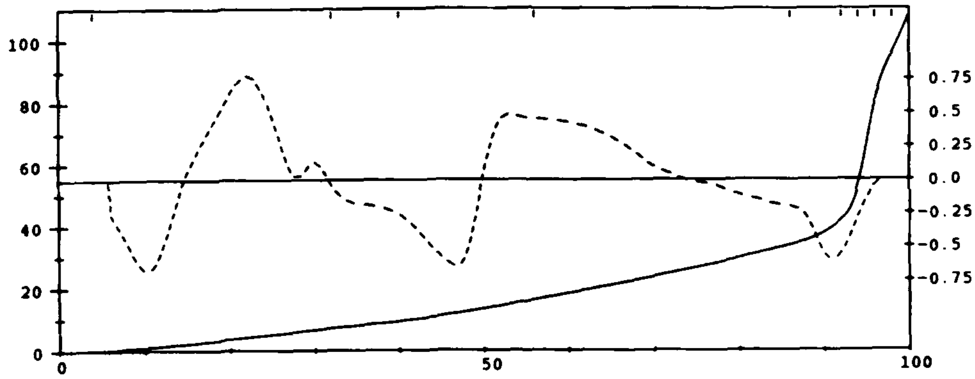


Figure 7. The final approximation with bound on the derivative after knot removal.

Figure 7 shows the result of applying constrained knot removal to f with the given tolerance $\epsilon = 1.0$. The dashed curve is the error function $g - f$, and the solid curve is the final spline approximation g to ϕ. The number of interior knots was reduced from 46 to 9, and the location of the remaining interior knots is marked at the top of the figure.

In Figure 8 the constraining function h (dashed curve) and g' (solid curve) are plotted. We note that the peak in g' is induced from the rapid increase in the data ϕ in that area.

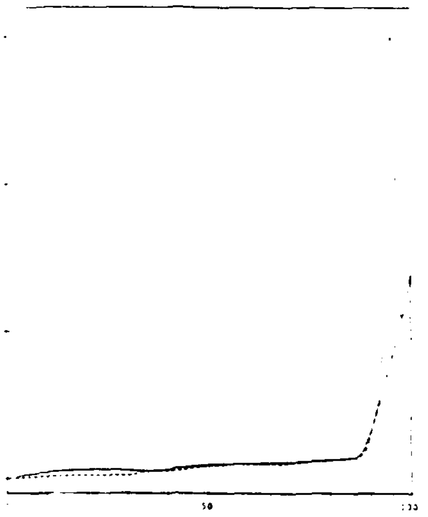


Figure 8. The first derivative of the final approximation and its lower bound.

References

de Boor, C. (1976a), Splines as linear combinations of B-splines. A survey. In *Approximation Theory II* (Lorentz, G. G., Chui, C. K., and Schumaker, L. L., Eds.), Academic Press, New York, pp. 1–47.

de Boor, C. (1976b), On local linear functionals which vanish at all B-splines but one. In *Theory of Approximation with Applications* (Law, A. G., and Sahney, B. N., Eds.), Academic Press, New York, pp. 120–145.

Cohen, E., and Schumaker, L. L. (1985), Rates of convergence of control polygons. *CAGD* **2**, 229–235.

Cox, M. G. (1987), Data approximation by splines in one and two independent variables. In *The State of the Art in Numerical Analysis*, (Iserles, A. and Powell, M. J. D., Eds.), Clarendon Press, Oxford, pp. 111–138.

Dahmen, W. (1986), Subdivision algorithms converge quadratically. *J. Comp. Appl. Math.* **16**, 145–158.

Ferguson, D. R., Frank, P. D., and Jones, A. K. (1988), Surface shape control using constrained optimization on the B-spline representation. *CAGD* **5**, 87–103.

Fletcher, R. (1987) *Practical Methods of Optimization*, Second edition, John Wiley & Sons, Chichester.

Fontanella, F. (1987), Shape preserving surface interpolation. In *Topics in Multivariate Approximation*, (Chui, C. K., Schumaker, L. L. and Utreras, F. I., Eds.), Academic Press, Orlando Florida, pp. 63–78.

Franke, R., and Schumaker L. L. (1987), A bibliography of multivariate approximation. In *Topics in Multivariate Approximation,* (Chui, C. K., Schumaker, L. L. and Utreras, F. I., Eds.), Academic Press, Orlando Florida, pp. 275–335.

Jia, R. Q. (1983), Total positivity of the discrete spline collocation matrix. *J. Approx. Theory* **39,** 11–23.

Lyche, T. (1988), Note on the Oslo Algorithm. *Computer Aided Design* **20,** 353–355.

Lyche, T., and Mørken, K. (1987a), A discrete approach to knot removal and degree reduction algorithms for splines. In *Algorithms for Approximation* (Mason, J. C. and Cox, M. G., Eds.), Clarendon Press, Oxford, pp. 67–82.

Lyche, T., and Mørken, K. (1987b), Knot removal for parametric B-spline curves and surfaces. *CAGD* 4, 217–230.

Lyche, T., and Mørken, K. (1988), A data-reduction strategy for splines with applications to the approximation of functions and data. *IMA J. Numer. Anal.* **8,** 185–208.

Utreras, F. I. (1987), Constrained surface construction. In *Topics in Multivariate Approximation,* (Chui, C. K., Schumaker, L. L. and Utreras, F. I., Eds.), Academic Press, Orlando Florida, pp. 233–254.

NEAR REAL-TIME SPLINE FITTING OF LONG SEQUENCES OF UNIFORMLY-SPACED DATA

G. T. ANTHONY AND M. G. COX
National Physical Laboratory, Teddington

Abstract Data is frequently captured in very long sequences at equal intervals in the independent variable. This situation occurs, for instance, in medical diagnosis where various quantities (eg intestinal pressure and acidity) are regularly recorded at short time intervals (seconds) over periods of many hours. There are consequent difficulties in storing and analysing the resulting very large amounts of data. Another difficulty in analysing the data (to discover peaks, zeros, etc) is that the data is often noisy. These difficulties are reduced by first fitting a smooth mathematical function to the data and then working instead with that function. We consider here an approximate least-squares fitting method using polynomial splines with uniformly spaced knots at some of the data points. Advantage is taken of the structure of the normal matrix to develop a technique that is near real-time in that it produces a "running fit" over successive blocks of data. The degree to which the method is approximate can be controlled. The algorithm is fast in operation and the fits have local properties.
Key words: B-splines, data approximation, least squares, medical data analysis

1. Introduction

Tijskens, Janssens, Vantrappen, De Bondt and Vandewalle [11], in considering the automatic analysis of medical data gathered from (digitised) multi-channel chart recorders, have presented a method for fitting extremely long sequences of uniformly spaced data. They seek a fit in the form of a cubic spline with prescribed uniform knots, because this function is sufficiently versatile to represent such data adequately. Moreover, this form is convenient for subsequent (medical) analysis of the tracings. Data reduction is important because it is necessary to avoid storing the very large amounts of data involved. This implies that the fitting should be performed essentially in "real time", that is as the data is gathered. To this end they use a "window" technique. Data in a window spanning an odd number of spline intervals is fitted by a spline in its B-spline form in the least-squares sense. using the singular value decomposition (SVD) [5, page 293 *et seq*]. The central coefficient of the fit is retained, the window advanced one interval and the process repeated. The window is chosen large enough so that, to the required accuracy, the coefficients obtained are identical to those that would have been calculated in a single fit to all the data.

The SVD is a powerful tool that, in this case, takes no advantage of the *structure* of the spline-fitting problem. Consequently, the computation time is unnecessarily long. Since the knots of the spline are prescribed, each B-spline coefficient is a linear combination of the data values in the window. The multipliers or weights in the linear combination can be pre-computed using an approach described below. Thus the method is reduced to the evaluation of a single inner product for each coefficient, with the same set of multipliers in each case. This paper describes this alternative approach.

2. Solution method

We shall let x denote the independent variable and y the dependent variable (in this case time and the medical response indicated on the chart recorder, respectively). Data values $\{y_r\}$ are assumed to be provided sequentially at corresponding points $\{x_r\}$ having a uniform spacing h. This data is to be approximated in near real-time by a polynomial spline $s(x)$ of order m (degree $m-1$) with uniformly spaced knots. A knot will be placed at every pth data point, and thus the knot spacing will be ph. It is algebraically convenient to scale the independent variable so that $ph = 1$. If this is done the scaled data spacing is p^{-1} and the knots are at the integers. p is then the data density, that is the number of data abscissae per unit of the independent variable x. Henceforth we shall work only in the scaled variables.

The spline s will be represented as a linear combination of B-splines,

$$s(x) = \sum_j c_j N_m(x-j),$$

where $N_m(x)$ is the normalized B-spline of order m with the knots $0, 1, \ldots, m$ [2].

A (linear) data fitting problem, in which the fitting function is expressed as a linear combination of basis functions (here the B-splines), can be formulated by writing the model equation for each data point, giving an overdetermined system of linear equations:

$$Ac = y. \tag{1}$$

In (1), each row of the observation matrix A contains values of the basis functions at a data point, y is the vector of corresponding data values and c is the vector of coefficients to be determined. The normal-equations approach [6, page 121 *et seq*] to obtaining the least-squares solution to (1) consists in forming and solving the equations

$$Hc = z, \tag{2}$$

where $H = A^TA$ and $z = A^Ty$.

The system of equations (2) has a unique solution and, equivalently, H is invertible if and only if A has full column rank.

The computation falls broadly into two parts: forming H and inverting it. In the present application it is particularly simple to form H because, with uniform knots, all B-splines (of a given order) are translations of one another. So the element in row r and column j of A is $N_m(x_r - j)$, and it follows from the compact support property of B-splines [2] and the uniform spacing of the data that H is a $(2m-1)$-diagonal Toeplitz matrix (that is the elements on each diagonal are constant). A row of H takes the form

$$\{\ldots,\ 0,\ h_{m-1},\ h_{m-2},\ \ldots,\ h_1,\ h_0,\ h_1,\ \ldots,\ h_{m-1},\ 0,\ \ldots\},$$

with

$$h_j = \sum_r N_m(x_r)N_m(x_r - j).$$

Because of the uniform knots and data, H is well conditioned and invertible for moderate values of m provided p is greater than unity. So the usual objection that the formation and solution of the normal equations can introduce errors that are greater in magnitude than those we expect with a stable algorithm is of no significance here.

Consider, as an example, the piecewise linear case, $m = 2$. Here

$$N_2(x) = \begin{cases} x, & x \in [0,\ 1), \\ 2 - x, & x \in [1,\ 2], \\ 0, & \text{otherwise}, \end{cases}$$

so that

$$\begin{aligned} h_1 &= \sum_r N_2(x_r)N_2(x_r - 1) \\ &= \sum_{r=p}^{2p} \left(2 - \frac{r}{p}\right)\left(\frac{r}{p} - 1\right) \\ &= \frac{1}{3!p}(p^2 - 1) \end{aligned}$$

and

$$\begin{aligned} h_0 &= \sum_r N_2(x_r)N_2(x_r) \\ &= \sum_{r=0}^{p} \left(\frac{r}{p}\right)^2 + \sum_{r=p}^{2p} \left(2 - \frac{r}{p}\right)^2 - 1 \\ &= \frac{1}{3!p}(4p^2 + 2). \end{aligned}$$

The typical row of H for B-splines of other orders can be obtained similarly, using the basic recurrence relation for B-splines [1] and formulae for the sums of powers of

Linear							
	$3!ph_0$	$=$	$4p^2$	$+$	2		
	$3!ph_1$	$=$	p^2	$-$	1		
Quadratic							
	$5!p^3h_0$	$=$	$66p^4$	$-$	6		
	$5!p^3h_1$	$=$	$26p^4$	$+$	4		
	$5!p^3h_2$	$=$	p^4	$-$	1		
Cubic							
	$7!p^5h_0$	$=$	$2416p^6$	$+$	$\frac{112}{3}p^2$	$+$	$\frac{200}{3}$
	$7!p^5h_1$	$=$	$1191p^6$	$-$	$21p^2$	$-$	$\frac{150}{3}$
	$7!p^5h_2$	$=$	$120p^6$	$+$		$+$	$\frac{60}{3}$
	$7!p^5h_3$	$=$	p^6	$+$	$\frac{7}{3}p^2$	$-$	$\frac{10}{3}$

TABLE 1

Expressions for the nonzero elements in the typical row of the normal matrix H.

the natural numbers, or by more sophisticated methods. The linear, quadratic and cubic cases are given in Table 1, where, for order m, each element is multiplied by $(2m-1)!p^{2m-3}$ giving even polynomials of degree $2m-2$.

Various properties of these polynomials are readily demonstrated. In particular, we note that the leading coefficients are the values of N_{2m} at the knots, the trailing coefficients are proportional to the binomial coefficients of order $2m-2$, and the coefficient of p^{2m-4} is zero (if $m > 2$). The first observation is analogous to the integral result [10]

$$\int_{-\infty}^{\infty} M_r(x-j)M_t(x-k)dx = M_{r+t}(j-k), \tag{3}$$

with $r = t$. In (3), $M_r = rN_r$ is the B-spline with integral normalization $\int_{-\infty}^{\infty} M_r(x)dx = 1$ [8].

The third observation means that (for $m > 2$) the leading term dominates by a factor $O(p^4)$. This factor gives an indication of the convergence of the solution of the *discrete* least-squares spline approximation problem, as considered here, to that of the *continuous* uniform-knot spline approximation problem of minimizing $\int_{-\infty}^{\infty}(y(x)-s(x))^2dx$.

The second part of the calculation, obtaining the inverse G of H, can also be variously accomplished. Yamamoto and Ikebe [12] give a direct formulation for the inversion of band matrices which is particularly simple for tridiagonal matrices. In our linear case, $m = 2$, this gives for the i,j-th element of G (symmetric)

$$g_{i,j} = -\frac{(\alpha^i - \beta^i)(\alpha^{n+1}\beta^j - \alpha^j\beta^{n+1})}{(\alpha-\beta)(\alpha^{n+1}-\beta^{n+1})}, \quad i \le j,$$

where α and β are the roots of

$$h_1t^2 + h_0t + h_1 = 0$$

and n is the dimension of H.

This is equivalent, for sufficiently large n, to a row

$$\{\ldots,\ g_2,\ g_1,\ g_0,\ g_1,\ g_2,\ \ldots\}$$

of G being given by

$$g_0 = \frac{1}{\sqrt{(p^2+2)/3}},$$

$$g_{j-1} = -kg_j - g_{j+1}, \quad j \geq 2, \quad k = h_0/h_1 = (4p^2+2)/(p^2+1).$$

The sequence $\{g_j\}$ can be calculated by setting $g_{N+1} = 0$, $g_N = 1$ for some (notional) N, and applying the recurrence above for decreasing values of the suffix until a g_{N-M} is calculated with a sufficient number of significant figures. The sequence is then renumbered and normalized so that g_{N-M} is $g_0 = 1/\sqrt{(p^2+2)/3}$.

The typical row of G for this order of spline is shown to 6 decimal places for p = 2, 4, 10 in the left-hand part of Table 2. For $p = 10$, the decay factor k is close to its lower limit of $2+\sqrt{3} \approx 3.732$.

The Yamamoto and Ikebe algorithm is less straightforward for $m > 2$. An alternative approach is to use software from any of the standard linear algebra algorithms libraries such as NAG [4] or LINPACK [3]. For demonstration purposes we used the INV function from PC-MATLAB [7]. This employs an LU decomposition of H and then inverts the factors. The typical (symmetric half-) rows of G for $m = 4$ (cubic) with $p = 2, 4, 10$ are shown in the right-hand part of Table 2. The decay factor is close to 2 in all three cases. In fact we can see from Table 1 that the decay tends to $2416/1191 \approx 2.0285$ as p increases.

Finally, the coefficients of the fit are obtained by multiplying an appropriate (symmetric) section of the typical row of G into the corresponding section of z. The choice of this section is equivalent to choosing the number of spline intervals in the window of the procedure of Tijskens *et al* [11]. It should be emphasised that once the order m and data density p have been chosen, the calculation of the typical row of G is only required once and can be pre-computed. This gives an *a priori* estimate of the number of data points in a window needed to achieve a given accuracy of fit.

A refinement of the method is to compute $X = GA^T$ (the pseudoinverse [6, page 36 *seq*] of A), either directly or as the product of G and A^T. The coefficients of the fit are then obtained as the scalar product of an appropriate part of a row of X and the corresponding part of y. Table 3 shows part of the (symmetric half-) rows of X for $m = 4$ and $p = 2, 4, 10$ produced using PC-MATLAB. The elements of X exhibit a "piecewise" decay, and it is again straightforward to determine the length of the section of a row of X (equal to the number of data points in a window) required to achieve a given accuracy.

In the above we have implicitly assumed that the range over which the response is measured (and fitted) is of infinite extent. It follows that the matrices A, H, etc are of infinite order. Obviously, the data is gathered on a finite range. Equally

$p = 2$	$p = 4$	$p = 10$
⋮	⋮	⋮
-0.000000	-0.000001	-0.000001
0.000000	0.000004	0.000003
-0.000003	-0.000018	-0.000015
0.000018	0.000078	0.000057
-0.000105	-0.000327	-0.000217
0.000612	0.001363	0.000824
-0.003571	-0.005672	-0.003132
0.020815	0.023595	0.011893
-0.121320	-0.098146	-0.045163
0.707106	0.408248	0.171498
-0.121320	-0.098146	-0.045163
0.020815	0.023595	0.011893
-0.003571	-0.005672	-0.003132
0.000612	0.001363	0.000824
-0.000105	-0.000327	-0.000217
0.000018	0.000078	0.000057
-0.000003	-0.000018	-0.000015
0.000000	0.000004	0.000003
-0.000000	-0.000001	-0.000001
⋮	⋮	⋮

$p = 2$	$p = 4$	$p = 10$
⋮	⋮	⋮
0.000001	0.000000	0.000000
-0.000001	-0.000001	0.000000
0.000002	0.000002	0.000001
-0.000005	-0.000003	-0.000001
0.000009	0.000006	0.000002
-0.000017	-0.000010	-0.000004
0.000032	0.000019	0.000008
-0.000060	-0.000036	-0.000015
0.000114	0.000068	0.000027
-0.000215	-0.000127	-0.000051
0.000407	0.000237	0.000095
-0.000768	-0.000443	-0.000178
0.001450	0.000828	0.000333
-0.002737	-0.001547	-0.000622
0.005168	0.002891	0.001162
-0.009758	-0.005403	-0.002170
0.018425	0.010098	0.004055
-0.034791	-0.018872	-0.007575
0.065690	0.035270	0.014151
-0.124025	-0.065913	-0.026434
0.234093	0.123144	0.049365
-0.441263	-0.229778	-0.092074
0.827044	0.426393	0.170791
-1.511359	-0.771993	-0.309097
2.445129	1.240342	0.496466
-1.511359	-0.771993	-0.309097
0.827044	0.426393	0.170791
⋮	⋮	⋮

Table 2

The left-hand part of the table gives the values of the elements in the typical row of G, the inverse of the normal matrix H, in the case $m = 2$ (spline of degree 1) for three values of p, the data density. The right-hand part is the counterpart of this in the case $m = 4$ (cubic spline), showing (symmetric) half-rows.

$p = 4$	$p = 10$
⋮	⋮
0.000005	-0.002426
0.000000	-0.002362
-0.000007	-0.002016
-0.000011	-0.001360
-0.000010	-0.000466
0.000000	0.000569
0.000013	0.001652
0.000021	0.002685
0.000018	0.003573
0.000000	0.004221
-0.000024	0.004533
-0.000040	0.004413
-0.000034	0.003766
0.000001	0.002541
0.000045	0.000872
0.000074	-0.001063
0.000063	-0.003085
-0.000001	-0.005015
-0.000084	-0.006675
-0.000138	-0.007886
-0.000118	-0.008469
0.000002	-0.008245
0.000157	-0.007037
0.000259	-0.004750
0.000220	-0.001632
-0.000005	0.001982
-0.000294	0.005758
-0.000484	0.009365
-0.000411	0.012466
0.000008	0.014730
0.000549	0.015823
0.000904	0.015410
0.000769	0.013159
-0.000016	0.008894
-0.001026	0.003079
-0.001690	-0.003666
continued ...	

$p = 2$	$p = 4$	$p = 10$
	... continued	
⋮	⋮	⋮
0.000030	0.002686	-0.029575
-0.000041	-0.000055	-0.028844
-0.000057	-0.003582	-0.024690
0.000078	-0.005902	-0.016789
0.000107	-0.005020	-0.005991
-0.000147	0.000103	0.006555
-0.000202	0.006695	0.019704
0.000278	0.011030	0.032307
0.000382	0.009383	0.043218
-0.000525	-0.000191	0.051290
-0.000722	-0.012511	0.055375
0.000992	-0.020614	0.054327
0.001363	-0.017540	0.046998
-0.001873	0.000347	0.032772
-0.002573	0.023371	0.013145
0.003536	0.038528	-0.009852
0.004859	0.032814	-0.034192
-0.006676	-0.000567	-0.057847
-0.009174	-0.043585	-0.078789
0.012606	-0.072007	-0.094988
0.017324	-0.061596	-0.104418
-0.023800	0.000387	-0.105050
-0.032720	0.080694	-0.094856
0.044916	0.134580	-0.072627
0.061847	0.117300	-0.040441
-0.084623	0.004759	-0.001194
-0.117319	-0.144547	0.042219
0.158243	-0.251472	0.086899
0.225926	-0.236873	0.129951
-0.286154	-0.053637	0.168479
-0.462210	0.217217	0.199584
0.433175	0.462635	0.220372
1.126300	0.569564	0.227945
0.433175	0.462635	0.220372
-0.462210	0.217217	0.199584
⋮	⋮	⋮

TABLE 3

The values of the elements in the typical (symmetric) half-row of X, the pseudoinverse of the observation matrix A, in the case $m = 4$ (cubic spline) for three values of p, the data density. Note the "piecewise decay" in the magnitude of the elements.

obviously, the observation matrix A is finite and not all its rows have the same nonzero elements, because the first few and last few rows are curtailed. These end effects lead to a matrix G whose first few and last few rows are not shifted copies of a typical row. However, for sufficiently large orders of matrix, the rows converge to a central section of the matrix whose rows do have this property to working precision. Furthermore, to that precision the typical row is the typical row of the infinite case. The required order is quite modest; the examples given above were computed with matrices of order 40.

3. Conclusion

We have described a near real-time algorithm for fitting arbitrarily long sequences of uniformly-spaced data. The method uses splines with uniform knots and achieves an approximation to the least-squares fit to all the data. If sufficient data values "local to" any particular B-spline coefficient in the approximant s are employed then, to any specified precision, s can be regarded as *the* least-squares fit. In particular, for practical purposes, the accuracy of the data indicates the number of data values to be employed. For instance, if the data density p is 10, the data is accurate to approximately two significant figures, and a cubic spline is used, each B-spline coefficient can be formed as a linear combination of about 150 of the data values. However, there is one "replacement" B-spline coefficient in this case for every 10 data values. Hence each data value is used in each of 15 B-spline coefficient evaluations. The total number of floating-point operations is therefore only 15 times the number of data points. Similar statements can be made for other values of m and p. The multipliers in the linear combination are formed by a straightforward pre-computation that depends on p and the data accuracy.

The case p equals 1 corresponds to placing a knot at every data point and hence to *interpolation*. Of course, this is a much-studied problem dating from the seminal work of Schoenberg [8, 9]. In one sense, the ideas here are a straightforward generalization of Schoenberg's work. Considerable work needs to be done, particularly on its theoretical aspects, to develop this generalization fully.

Acknowledgments

We thank our colleagues Alistair Forbes and Peter Harris for useful discussions and for reading a draft of this paper.

References

1. M. G. Cox. The numerical evaluation of B-splines. *J. Inst. Math. Appl.*, 10:134–149, 1972.
2. M. G. Cox. Practical spline approximation. In P. R. Turner, editor, *Notes in Mathematics 965: Topics in Numerical Anlaysis*, pages 79–112, Berlin, 1982. Springer-Verlag.
3. J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart. *LINPACK User's Guide*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1979.
4. B. Ford, J. Bently, J. J. du Croz, and S. J. Hague. The NAG Library 'machine'. *Software - Practice and Experience*, 9:56–72, 1979.
5. G. H. Golub and C. F. Van Loan. *Matrix Computations*. North Oxford Academic, Oxford, 1983.
6. C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.
7. C. Moler, J. Little, and S. Bangert. *PC-MATLAB*. The MathWorks, Inc., Sherborn, Ma, USA, 1987.
8. I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. Part A–On the problem of smoothing or graduation. A first class of analytic approximation formulae. *Qu. Appl. Math.*, 4:45–99, 1946.
9. I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. Part B–On the problem of osculatory interpolation. A second class of analytic approximation formulae. *Qu. Appl. Math.*, 4:112–141, 1946.
10. I. J. Schoenberg. Cardinal interpolation and spline functions. *J. Approx. Theory*, 2:167–206, 1969.
11. G. Tijskens, J. Janssens, G. Vantrappen, F. De Bondt, and J. Vandewalle. Spline functions used for on-line data reduction and analysis of motility tracings. Presented at the Contact Group on Numerical Analysis, Splines and their Applications in Modelling, Namur, Belgium, 4 December, 1987.
12. T. Yamamoto and Y. Ikebe. Inversion of band matrices. *Linear Alg. Appl.*, 24:105–111, 1979.

AN ALGORITHM FOR KNOT LOCATION IN BIVARIATE LEAST SQUARES SPLINE APPROXIMATION

M BOZZINI
Dipartimento di Matematica
Università di Lecce
F DE TISI
Dipartimento di Matematica
Università di Milano

Abstract An automatic algorithm for the determination of a lattice suitable for the construction of bivariate least squares splines is studied. Some numerical examples are quoted.

Key words: Bivariate splines, Knot location, Least squares approximation.

1. Introduction

The problem of constructing a surface $F(x,y)$, $(x,y) \in D \subset \mathcal{R}^2$, from a discrete number of points (x_i, y_i, f_i) $(i = 1, \ldots, N)$ has been studied by many authors. In the literature there appears to be greater concentration on the development of algorithms for data interpolation than for approximation of data subject to errors, where we are faced with additional difficulties.

When spline functions are employed, it is well-known that cubic splines afford a good balance between efficiency and goodness of fit. These functions are generally written in terms of the tensor product of B-spline basis functions $\{B_\ell(x)\}$ with $\{B_m(y)\}$ defined on the Cartesian axes.

In the case of cubic spline interpolation, an interpolant to data on any rectangular grid can be constructed. If the grid is defined by the lines $x = x_i$ $(i = 1, \ldots, N_x)$, $y = y_j$ $(j = 1, \ldots, N_y)$, then the points x_i and y_j determine the knots for the interpolating spline.

When the data values contain random errors, we may construct an approximation by the method of least squares. The approximating function is written in terms of basis functions, fewer in number than the data points, defined on a knot set which is different from the original data point set. The choice of knots can greatly affect how well the surface fits the data.

In this paper an algorithm is presented for the automatic placement of the knots in the case of bivariate least squares spline approximation. Before describing this algorithm, we briefly review in the next section how to construct such approximations.

2. Bivariate least squares spline approximation

Consider a rectangle $R = \{(x,y) \in \mathcal{R}^2 : x \in [a,b], y \in [c,d]\}$ and the lattice defined by the knots $\{\lambda_i\}_{i=0}^{h+1}$ and $\{\mu_j\}_{j=0}^{g+1}$, where

$$a = \lambda_0 < \lambda_1 < \ldots < \lambda_{h+1} = b$$

and

$$c = \mu_0 < \mu_1 < \ldots < \mu_{g+1} = d.$$

A bivariate spline of order k on R with interior knots $\{\lambda_i\}_{i=1}^{h}$ and $\{\mu_j\}_{j=1}^{g}$ is a function $s(x,y)$ with the properties that:

i) On the subrectangle $R_{ij} = [\lambda_{i-1}, \lambda_i] \times [\mu_{j-1}, \mu_j]$ $(i = 1, \ldots, h+1;\ j = 1, \ldots, g+1)$, $s(x,y)$ is a polynomial of total order $2k$ in x and y.

ii) The functions

$$\frac{\partial^{p+q}}{\partial x^p \partial y^q} s(x,y) \qquad (0 \le p, q \le k-2)$$

are continuous on R.

A generally well-conditioned basis for $s(x,y)$ is given by the tensor product of k^{th} order univariate B-splines $M_i(x)$ with knots λ_u $(u = 1, \ldots, h)$ and $N_j(y)$ with knots μ_v $(v = 1, \ldots, g)$:

$$s(x,y) = \sum_{i=1}^{h+k} \sum_{j=1}^{g+k} \nu_{ij} M_i(x) N_j(y).$$

The least squares spline fit to the given data is the solution of the problem

$$\min_{\nu_{ij}} \sum_{r=1}^{N} \left(f_r - \sum_{i,j} \nu_{ij} M_i(x_r) N_j(y_r) \right)^2 .$$

Equivalently, we may solve, in the least squares sense, the overdetermined system of equations

$$A\nu = f, \tag{1}$$

where $\boldsymbol{\nu} = \{\nu_{ij}\}_1^{h+k,g+k}$, expressed as a $(h+k)(g+k)$-vector, $\boldsymbol{f} = \{f_r\}_1^N$ and A is the $N \times (h+k)(g+k)$ matrix whose r^{th} row contains the basis function values $M_i(x_r)N_j(y_r)$, with

$$N > (h+k)(g+k). \tag{2}$$

A solution can be found using a numerically stable method which employs Householder transformations. These reduce the matrix A to triangular form by means of orthogonal transformations. Another method is based on Givens rotations. These are particularly useful in the case of structured matrices, are numerically stable and reduce the storage requirements.

The major difficulty that occurs in determining the least-squares solution of (1) is the possible rank deficiency of the matrix A which depends on the lattice used. In the univariate case, A has full rank if the Schoenberg-Whitney conditions are satisfied (Cox, 1986; de Boor, 1978), while in the bivariate case conditions exist only for particular configurations of the points and knots. Rank deficient matrices occur very frequently and lead to non-unique solutions. We can determine a particular solution selecting either the minimum norm solution (which, however, is not invariant under translation) or a geometrically invariant solution (Cox, 1986).

3. Knot placement

From what we have said above, a need clearly arises for subdividing the rectangle R into a lattice that affords a good balance between the closeness of the fit to the data and the smoothness of the fit, whilst reducing the number of empty panels.

In the literature we find results for data on a mesh and scattered data. In the former case an algorithm due to Dierckx (1981) can be used. This algorithm constructs a lattice in such a way as to control the above-mentioned balance.

When the data are scattered over the domain, the literature suggests methods which are based on sequences of "trials". More precisely, one forms an initial rectangular lattice and constructs the relevant spline; if the fit is not satisfactory in some regions, the lattice is refined appropriately (see Cox (1986)).

4. The proposed algorithm

Our objective is to construct an algorithm which determines a suitable lattice for any point configuration. In order that the behaviour of the fitted surface is sensible and in particular sufficiently smooth, our algorithm is required

i) to provide a subdivision which achieves near-uniformity in the point distribution between the mesh lines,

ii) to impose a limitation on the knot spacing.

The limitation in ii) is controlled by an index L, as described below.

We first determine a functional relationship between L and the number of data points N. To this end, we consider the real variable n, which coincides with N at the points of integer abscissa, and require that the following conditions are satisfied:

$$\begin{aligned}
\lim_{n\to 0} L &= \infty, \\
\lim_{n\to\infty} L &= \text{constant}, \\
L &> 0 \quad \forall\, n \in \mathcal{R}^+, \\
L' &< 0 \quad \forall\, n \in \mathcal{R}^+.
\end{aligned}$$

As a first approximation, we write

$$L = \rho \frac{1}{n^\alpha} + \gamma, \tag{3}$$

where β and γ are positive constants which depend on the measure $\mu(R)$ of the domain, and $\alpha \in \mathcal{R}^+$. The relationship (3) must also satisfy the following two conditions:

1) The decrement in L must not in magnitude be much larger than the relevant increment in N.
2) When the given points are dense in the domain, L must be nearly insensitive to variations in N.

If we consider a value n, an increment Δn in n and the corresponding values $L(n)$ and $L(n + \Delta n)$, we have

$$\Delta L = L(n + \Delta n) - L(n) = L'(n)\Delta n + R_2(\Delta n),$$

where $R_2(\Delta n)$ is the remainder from the Taylor formula. From this, for Δn sufficiently small,

$$\Delta L = L'(n)\Delta n = -\frac{\alpha\beta}{n^{\alpha+1}}\Delta n.$$

In order to satisfy the first of the above conditions, it is necessary that $0 < \alpha \leq 1$; whereas, for the second, it is convenient to take $\alpha \geq 1$. As a consequence, a suitable value is $\alpha = 1$, giving

$$L = \beta \frac{1}{n} + \gamma. \tag{4}$$

This conjecture is corroborated by practice. Therefore, when forming the lattice, we shall consider (4) and the first assumption i). In order to satisfy the latter, we proceed in the following way.

Let us think of the points as point masses and determine their barycentre. If the points are regularly distributed, the barycentre is the centre of symmetry of the system and its coordinates subdivide the domain into four equal and homogeneous parts. Otherwise, the barycentre lies in the zone where the density is the greatest;

in this case, the subdivision makes the four zones obtained more homogeneous than any other subdivision.

The algorithm works in the following way. For a given set I_N of points $P_i = P(x_i, y_i)$ $(i = 1, \ldots, N)$, we determine the point $P^{(0)} \in I_N$ which has the minimum Euclidean distance from the barycentre B. We then find a circular neighbourhood $U(P^{(0)})$ about $P^{(0)}$ with radius r depending on the mean density of the data in the domain R:

$$r = k\frac{\mu(R)}{N}.$$

The factor k depends on the point configuration. If we have some information on the lattice, the user can assign an initial value k^*. Otherwise, the following rule can be used.

Suppose the N data points are placed on a rectangular grid as follows:

$$a = x_1 < x_2 < \ldots < x_{N_x} = b,$$

$$c = y_1 < y_2 < \ldots < y_{N_y} = d,$$

and define

$$h_1 = x_{i+1} - x_i = (b - a)/(N_x - 1),$$

$$h_2 = y_{i+1} - y_i = (d - c)/(N_y - 1).$$

In order that the neighbourhoods cover the domain, it is sufficient to choose the radius such that

$$r \geq \sqrt{h_1^2 + h_2^2}.$$

Hence,

$$k \geq \frac{N}{\mu(R)}\sqrt{\frac{(b-a)^2}{(N_x - 1)^2} + \frac{(d-c)^2}{(N_y - 1)^2}}. \qquad (5)$$

Thus, we shall thus choose an initial value k^* for k satisfying (5).

Once we have determined $U(P^{(0)})$, we calculate the barycentre B_0 of the points $P_i \in I_N \cap U(P^{(0)})$ and call (x_{B_0}, y_{B_0}) the coordinates of B_0. We proceed by considering the set of points $P_j \in (I_N - U(P^{(0)})) = I_{N_1}$; the algorithm determines the neighbourhood $U(P^{(1)})$, with radius r, of the point $P^{(1)} \in I_{N_1}$ which has the minimum distance from the barycentre B. Then we calculate, as above, the barycentre B_1 of the points $P_i \in I_N \cap U(P^{(1)})$. This procedure is repeated until there are no more points P_i remaining. If, for some $P^{(*)}$, we have $U(P^{(*)}) = \emptyset$, this point is not taken into consideration, and we proceed to determine the next point closest to B.

When the procedure ends, we see if (2) is satisified; if not, we repeat the procedure iteratively by incrementing the value of k until (2) is satisfied.

The set A_{B_j} of the coordinates (x_{B_j}, y_{B_j}) of the barycentres B_j $(j = 0, \dots, m)$ can be considered as a potential set of knots on the relevant coordinate axes, defining a first lattice. At this point, the algorithm controls the knot spacing by considering the sequences $\{x_{B_0}, \dots, x_{B_m}\}$ and $\{y_{B_0}, \dots, y_{B_m}\}$ of coordinates. For any couple B_j, B_{j+1} consider the inequalities:

$$|x_{B_{j+1}} - x_{B_j}| > L, \tag{6}$$

$$|y_{B_{j+1}} - y_{B_j}| > L. \tag{7}$$

If (6) and (7) are satisfied, the knots are accepted. Otherwise, if one of the above relationships is not verified for at least one of the couples B_j and B_{j+1}, we proceed as follows.

Suppose (6) is not verified. We consider the knots

$$x_{B_{j-1}},\ x_{B_j},\ x_{B_{j+1}} \text{ and } x_{B_{j+2}},$$

and define $\overline{x}_j$, $\overline{x}_{j+1}$ and $\overline{x}_{j+2}$ to be the midpoints of the intervals $[x_{B_l}, x_{B_{l+1}}]$ $(l = j-1, \dots, j+2)$. If $\overline{N}_{x_j}$ and $\overline{N}_{x_{j+1}}$ are, respectively, the number of points (each with its multiplicity) belonging to the intervals $[\overline{x}_j, \overline{x}_{j+1}]$ and $[\overline{x}_{j+1}, \overline{x}_{j+2}]$, we determine in the interval $[x_{B_j}, x_{B_{j+1}}]$ the point x^* for which the equality

$$x^*\overline{N}_{x_j} = (x_{B_{j+1}} - x_{B_j} - x^*)\overline{N}_{x_{j+1}}$$

holds. The sequence of knots on the x-axis is then modified to

$$x_{B_0}, \dots, x_{B_{j-1}},\ x^*,\ x_{B_{j+2}}, \dots, x_{B_m}.$$

5. Numerical results

In this section we use an example found in the literature, namely the test function $F(x, y) = \exp(-x^2 - y^2)$. The function is assumed specified by a sample of values

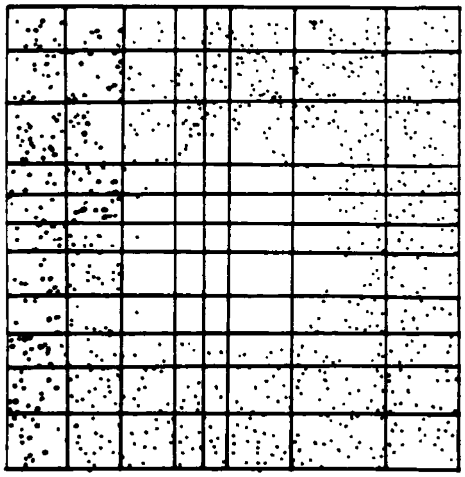
$$f_i = F_i + \epsilon_i, \qquad \epsilon_i \in U[-0.5 \times 10^{-3}, 0.5 \times 10^{3}],$$

where U is the uniform distribution and the approximating spline is that of minimum norm.

The error indices we have considered are

$$\text{L.S.E.} = \sqrt{\sum e_i^2/N}, \qquad \text{A.E.} = \sum |e_i|/N \ \text{ and } \ \text{MAX} = \max_{1 \le i \le N} |e_i|,$$

where $e_i = F(x_i, y_i) - s(x_i, y_i)$.



Example 1

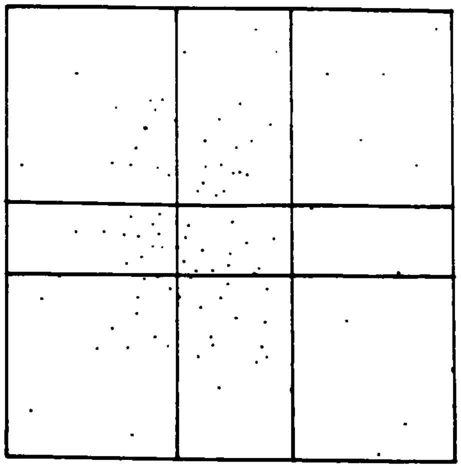
$R = [0,1] \times [0,1]$

$N = 1000$

$\text{L.S.E.} = .1179E - 03$

$\text{A.E} = .9284E - 04$

$\text{MAX} = .4642E - 03$



Example 2

$R = [-1,1.5] \times [-1,1]$

$N = 100$

$\text{L.S.E.} = .1091E - 02$

$\text{A.E.} = .8577E - 03$

$\text{MAX} = .3078E - 02$

6. Concluding remarks

For small data sets, the algorithm presented provides good results both for data on a mesh and for scattered data. For large data sets, our results compare favourably with those provided by methods discussed in the literature. The comparison holds also for computation time. Moreover, memory requirements are not excessive, so the algorithm can be used on a personal computer and can be modified for interactive use.

References

1. M.G. Cox. Data approximation by splines in one and two independent variables. Technical Report DITC 77/86, National Physical Laboratory, 1986.
2. C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.
3. P. Dierckx. An algorithm for surface fitting with spline functions. *I.M.A. J. Num. Anal.*, 1:267–283, 1981.

A KNOT PLACEMENT STRATEGY FOR LEAST SQUARES SPLINE FITTING BASED ON THE USE OF LOCAL POLYNOMIAL APPROXIMATIONS

M. G. COX, P. M. HARRIS and Helen M. JONES
National Physical Laboratory

Abstract We are concerned with the use of univariate spline functions in fitting noisy data in the least-squares sense. When defining the linear space of approximating functions from which our fit is to be taken, there is freedom both in the number and locations of the interior knots. The distribution of these knots can have a profound influence on how well the spline fits the data.

We describe an algorithm for determining an initial estimate of the number of knots, and their distribution, from which an initial fit to the data is obtained. The algorithm exploits in a natural way the piecewise polynomial nature of the approximating function by isolating subsets of the data which are adequately described by simple polynomials. Once an initial set of knots is known, adaptive knot placement strategies may be used to improve this set.

Key words: Knot placement, Least squares, Local polynomial, Polynomial spline.

1. Introduction

Polynomial spline functions are widely used for fitting "noisy" data; that is, data in which there are random errors in the dependent variable values. In practice, such data can arise when these values are measurements of the response of a physical system to known discrete inputs (the independent variable values). Using the method of *least squares*, a polynomial spline approximation to the underlying function represented by the data can be constructed.

When defining the linear space of splines from which the fit is to be taken, there is considerable freedom of choice in both the order n of the spline and in the number and locations of the interior knots λ_j, $j = 1, \ldots, N$. We are interested in constructing strategies and algorithms for automatically choosing $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$ so that the spline fit of given order with these knots is, in some sense, an "acceptable" fit to the data.

In Cox, Harris and Jones (1987) we discuss various approaches to solving the problem of automatic knot placement. Algorithms for deriving an *initial* trial set of knots and *adaptive* knot placement strategies are both considered. In particular, we present there a new version of an algorithm given in de Boor (1978) which we summarise below.

Let the data consist of the points (x_i, f_i), $i = 1, \ldots, m$, where $x_i \in [x_{min}, x_{max}]$ for all i. Suppose that we require a spline fit of order n (degree at most $n-1$) to the data and we wish to position N interior knots in (x_{min}, x_{max}). We first define *exterior* knots $\lambda_j = x_{min}$ for $j < 1$, $\lambda_j = x_{max}$ for $j > N$ and then, according to this strategy, we try to choose λ_j, $j = 1, \ldots, N$, such that

$$\int_{\lambda_k}^{\lambda_{k+1}} |f^{(n)}(x)|^{1/n}\,dx = \frac{1}{N+1}\int_{x_{min}}^{x_{max}} |f^{(n)}(x)|^{1/n}\,dx, \qquad (1)$$

for $k = 0, \ldots, N$. We assume that f, the (unknown) underlying function, belongs to the continuity class $C^n[x_{min}, x_{max}]$. The criterion (1) gives an interior knot distribution which is asymptotically equivalent, as $N \to \infty$ with $\max_{k=0,\ldots,N} |\lambda_{k+1} - \lambda_k| \to 0$, to that obtained from solving

$$\min_{\boldsymbol{\lambda}} \left\{ \max_{k=0,\ldots,N} \text{dist}_{[\lambda_k, \lambda_{k+1}]}(f, S^n_{\boldsymbol{\lambda}}) \right\}. \qquad (2)$$

In (2), "$\text{dist}_{[\lambda_k,\lambda_{k+1}]}(f, S^n_{\boldsymbol{\lambda}})$" represents the "distance", measured in the infinity norm over $[\lambda_k, \lambda_{k+1}]$, between f and $S^n_{\boldsymbol{\lambda}}$, the linear space of splines of order n with interior knot vector $\boldsymbol{\lambda}$. Criterion (1) is derived from (2) by replacing this distance function by a suitable bound.

Our version of the algorithm differs from de Boor's in that we approximate $f^{(n)}$ by a piecewise linear function rather than by a piecewise constant function. As a result we hope to achieve a better, and certainly smoother, approximation. The strategy is adaptive since the approximation to $f^{(n)}$ is obtained from a spline fit to the data based on a *current* set of interior knots. Thus, one application of the criterion results in this current set being updated, and the new knot vector forms the current knots at the next application.

In order to use (1) adaptively, we need

(a) an initial interior knot vector $\boldsymbol{\lambda}$,
(b) a means of approximating $f^{(n)}$ using the current set of knots, and
(c) a means of terminating the algorithm.

Details of how the approximation to $f^{(n)}$ is constructed is given in Cox et al (1987). Observe that (a) will need to be found *independently* of a current knot set since, to begin with, no such set will exist. Moreover, the effectiveness of this algorithm will depend on the quality of the initial approximation.

In this paper we present an algorithm for determining initial values for N and λ_j, $j = 1, \ldots, N$. The idea is to try to deduce information about the underlying function f from simple polynomial fits to subsets of the complete data set. These "local" fits give us "local" information about f independently of a set of interior knots. We then use this information to construct λ. We assume throughout that there is enough data to ensure that proper subsets which adequately describe f exist. Once an initial set of knots is known, the adaptive knot placement strategy described above may be used to improve this set.

In Section 2 we present an algorithm which associates a local data set and a corresponding local polynomial fit with each point "sufficiently interior" to the complete data set. We show in Section 3 how this local information may be used in a strategy for knot placement. Results of using the strategy are also presented in this section. Finally, in Section 4 we summarise the work.

2. Deriving Local Information About f

Given m data points $(x_i, f_i), i = 1, \ldots, m$, with $x_1 < \ldots < x_m$, representing an unknown underlying function f, we show how we may associate with each data point *sufficiently interior* (see later) a *local data set* and a *local polynomial fit* of order n.

Consider the i^{th} point (x_i, f_i). We first construct a subset of contiguous points taken from the complete data set, with indices centred on i, and consisting of $n+1$ points for n even or n points for n odd. We regard this subset as *local* to the i^{th} point. It is assumed that these points can be "adequately" fitted by a simple polynomial of order n. This will certainly be true in the case when n is odd since we can *interpolate* the local data by such a polynomial. It is now also evident what we mean by a point being sufficiently interior - it is a point for which this initial local data set can be constructed.

Our next step is to increase the size of the subset by adding the neighbouring point at each end if this can be done. A polynomial fit of order n to the new local data is computed and Powell's trend test (Powell, 1970) is used to assess the "acceptability" of the fit.

The largest subset so formed for which the polynomial fit to this data satisfies the test is taken to be *the local data set* associated with the i^{th} point. The procedure is then repeated for all sufficiently interior points.

In the use of the algorithm as described, there can be a problem, particularly for large data sets, with regard to the time taken for its execution. This is because we may need to compute a large number of polynomial fits (and evaluate these at every point of the corresponding data set for use in the trend test) for each of a large number of points.

To reduce the amount of work we use the size of the final local data set associated with the $(i-1)^{th}$ data point as a starting value for the size of the local data set for the i^{th} point. This set is then enlarged or reduced depending on whether it can adequately be fitted by a polynomial in the sense described. It is not clear whether we can conclude that if the fit to this set of data satisfies the trend test then the fits to all smaller subsets must also satisfy the test. But equally, if we always start with the smallest subset and look for the first local data set to fail the trend test, it is not clear that all larger sets must also fail the test. Thus, we believe that reducing the work in this way is not detrimental to the results produced by the algorithm.

The execution time is also affected by the manner in which the polynomial fits are computed. Traditionally, the methods used to generate a least-squares polynomial fit to data are those of Forsythe (Forsythe, 1957) and Forsythe-Clenshaw (Clenshaw, 1960). We use Forsythe's method in preference to that of Forsythe-Clenshaw because for our purposes it is appreciably faster.

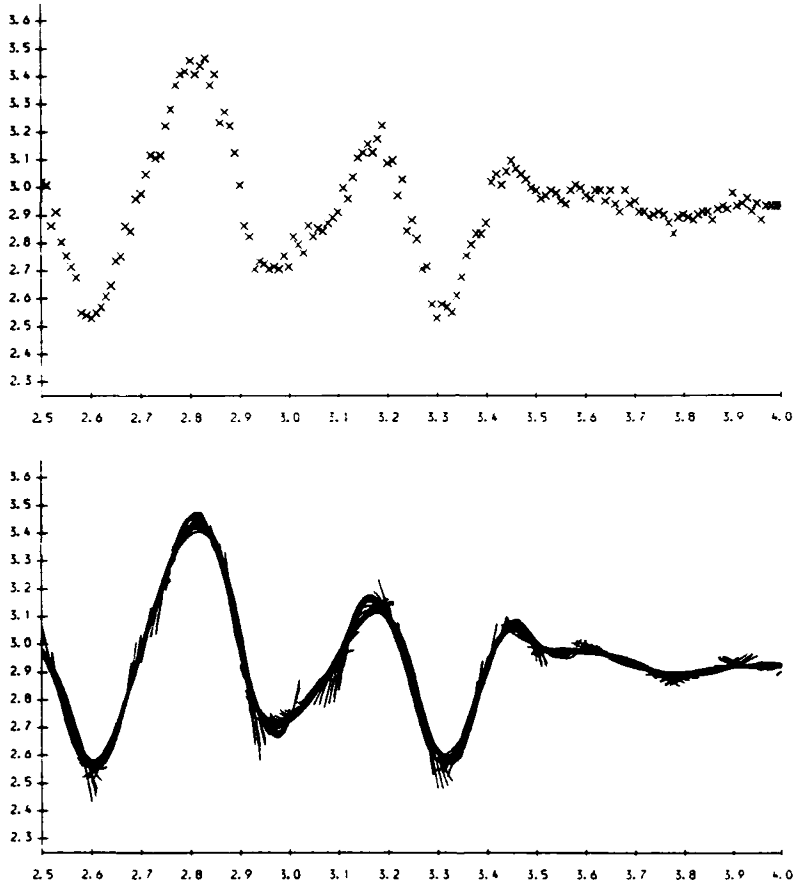


FIG. 1. *A subset of the data obtained from measurements of the response of an oscilloscope to a step input function and local polynomial fits obtained using a trend test.*

Figure 1 shows the local polynomial fits as produced by the algorithm for a set of data generated by measuring the response of an oscilloscope to a step input function. The complete data set consists of 512 points with equispaced abscissa values but we show here only a subset of the data in order to illustrate more clearly the local polynomials. The first diagram shows this subset and the second the local polynomial fits of order 4 associated with these and neighbouring points. We notice that the aggregate of the simple polynomial fits appears to be adequately describing the underlying function. In the following section we use the complete data set in our knot placement strategy.

3. An Algorithm for Knot Placement

In this section we show how we may use the information generated by the algorithm presented in Section 2 to determine an initial trial set of knots. These knots are used in a least squares spline fit to the data, and may be subsequently updated according to (1) to give improved fits.

Suppose we compute the local data sets associated with all sufficiently interior data points using polynomial fits of order n. Let the first and last data point in each local data set mark the *lower extent* and *upper extent*, respectively, of that set. Then, we position a knot at any point which is the lower (or upper) extent of more than two local data sets associated with consecutive data points. In this way we derive not only an estimate for N but also for the initial knot positions λ_j, $j = 1, \ldots, N$.

If the local data set for the i^{th} point is determined independently of those for the other points, then this strategy clearly gives a vector $\boldsymbol{\lambda}$ whose elements do not depend on the order in which the points are processed. However, we observed in Section 2 that for reasons of efficiency, we use the size of the final local data set for the $(i-1)^{th}$ point as a starting value for the size of the local data set for the i^{th} point. Thus, the points are necessarily processed from left to right and there is a dependence between the local data sets for consecutive data points. A better criterion is to locate a knot at an extent of a local data set which is *greater than or equal to* the extents of more than two subsequent and consecutive local data sets. Similarly, if the points were processed from right to left, we would use a "less than or equal to" test here.

Figure 2(a) shows the result of applying the knot placement strategy to the oscilloscope data, part of which is illustrated in Figure 1. The data points (small crosses) are shown together with the knot positions (vertical bars) and the fourth order spline fit to the data with these knots. There are 70 knots and the root mean square residual in the fit is 0.0543. We notice that in some regions there are *clusters* of knots and these can cause *overfitting* of the data. We say that overfitting occurs when the spline fit passes so close to the data that it begins to model the noise.

To remove the clusters we use a simple strategy which requires that each knot interval contains at least n ($= 4$, in this example) data points. If there are fewer than n points in an interval, the right-hand knot is removed (excepting the last knot interval where it is the left-hand knot which is removed). As a result, we are left with 44 knots and a spline fit as shown in Figure 2(b). The root mean square residual in this fit is 0.0605. The fit looks much more acceptable than that of Figure 2(a) because the oscillations in the spline that were previously present have been eliminated or reduced.

Finally, if we use these 44 knots to initialize the adaptive knot placement algorithm (1), and continue to remove clusters of knots as they are generated, we produce after 4 iterations the knots and associated spline fit shown in Figure 2(c). There are 42 knots here and the root mean square residual in the fit is 0.0618. Indeed, the use of (1) has produced little improvement over the set of 44 knots. We note, however, that the distribution of the reduced set of knots is markedly different.

Figure 3 shows the results at the same stages for another practical set of data.

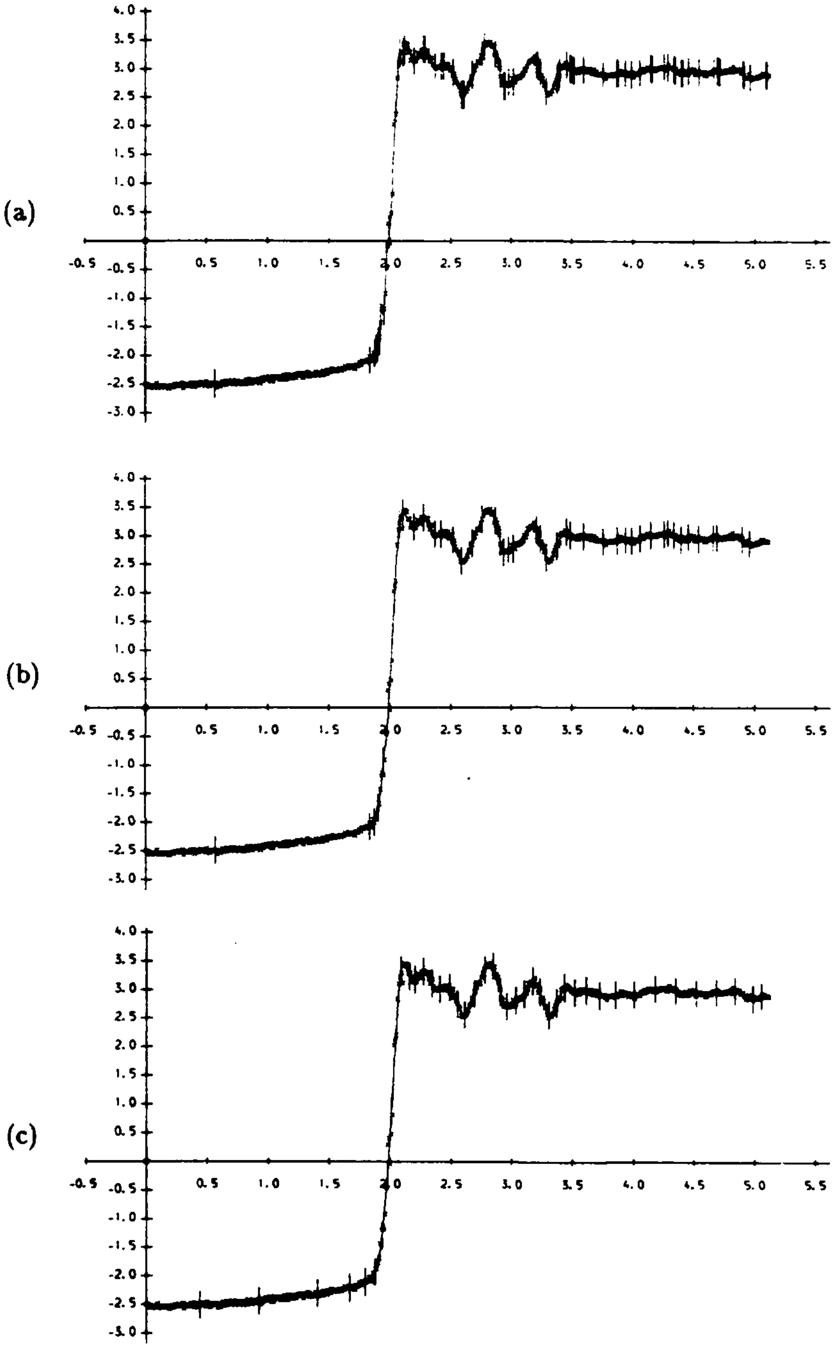


FIG. 2. ***Automatic knot placement applied to oscilloscope data:***
(a) Initial knots (70 knots, r.m.s. residual =* 0.0543*).
(b) After clusters are removed (44 knots, r.m.s. residual =* 0.0605*).
(c) After adaptive knot placement (42 knots, r.m.s. residual =* 0.0618*).

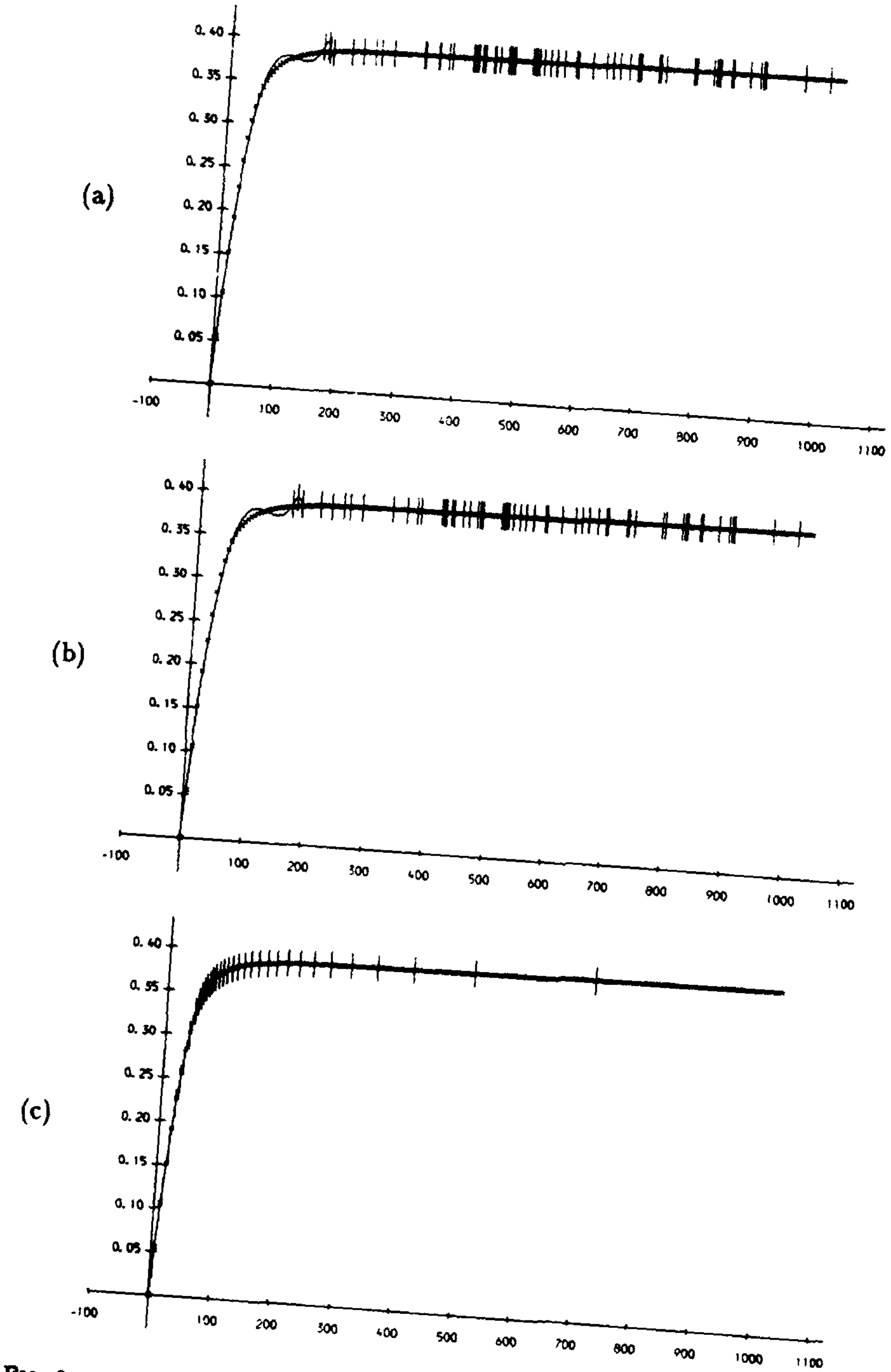


FIG. 3. *Automatic knot placement applied to photodiode data:*
(a) Initial knots (79 knots, r.m.s. residual = 0.00280).
(b) After clusters are removed (55 knots, r.m.s. residual = 0.00295).
(c) After adaptive knot placement (34 knots, r.m.s. residual = 2.73×10^{-7}).

This arises from very accurate measurements of the response of a photodiode. The data set consists of 1024 points with equispaced x_i-values. For clarity, we have plotted only every fifth point in the diagrams of Figure 3. A set of 79 knots is initially generated and this is reduced to 55 when clusters are removed as described above. The corresponding fits are illustrated in Figures 3(a) and (b) and the root mean square residuals are 0.00280 and 0.00295, respectively. In this example, the use of (1) gives a new set of knots which results in a very substantial improvement in the root mean square residual. Figure 3(c) shows the knots and corresponding spline fit after 4 iterations of adaptive knot placement. There are 34 knots here, and the root mean square residual in the fit is 2.73×10^{-7}. Adaptive knot placement has reduced the root mean square residual by a factor of 10,000 without overfitting. We observe that much of this improvement results from the strategy repositioning the knots in such a way as to remove the severe oscillation that previously existed in the "shoulder" of the curve.

4. Summary

In this paper we have presented a strategy for automatically deriving a knot vector $\boldsymbol{\lambda}$ to be used in constructing a least squares spline fit of given order to noisy data. We determine subsets of the complete data set which are adequately fitted by simple polynomials. These polynomials provide local information about the underlying function which the data represents. By considering how these subsets, or local data sets, interact we generate initial estimates for N and λ_j, $j = 1, \ldots, N$. With these initial values we may then use an adaptive knot placement algorithm to produce improved knot distributions. The particular adaptive strategy we use here relocates knots according to the criterion (1) and removes knots to prevent clustering. The examples we have given illustrate some of the features of our approach. For further details and examples, see Cox, Harris and Jones (1988).

Acknowledgement
We would like to thank Mr. G. T. Anthony for his comments on an earlier draft of this paper.

References

1. C.W. Clenshaw. Curve fitting with a digital computer. *Comp. J.*, 2:170–173, 1960.
2. M.G. Cox, P.M. Harris, and H.M. Jones. Strategies for knot placement in least squares data fitting by splines. Technical Report DITC 101/87, National Physical Laboratory, 1987.
3. M.G. Cox, P.M. Harris, and H.M. Jones. A knot placement strategy for least squares spline fitting based on the use of local polynomial approximations. National Physical Laboratory Technical Report, in preparation, 1988.

4. C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.

5. G.E. Forsythe. Generation and use of orthogonal polynomials for data fitting with a digital computer. *J. Soc. Indust. Appl. Math.*, 5:74–88, 1957.

6. M.J.D. Powell. Curve fitting by splines in one variable. In J.G. Hayes, editor, *Numerical Approximation to Functions and Data*, pages 65–83. The Athlone Press, London, 1970.

AN ALGORITHM FOR NONLINEAR SPLINES WITH NON-NEGATIVITY CONSTRAINTS

G. OPFER
Institute of Applied Mathematics
University of Hamburg

Abstract The problem of minimizing the exact form of the strain energy of a "spline" subject to non-negativity constraints is treated. Several necessary optimality conditions in terms of systems of differential equations are derived. Due to the constraints, additional boundary conditions have to be introduced at points which are not known in advance. An algorithms is outlined, which we call a *local* algorithm, which is based on only two grid points.

Key words: Curvature functional, Elastica with constraints, Global spline algorithm, Local spline problem, Nonlinear splines, Non-negativity constraints, Obstacle problem, Splines with constraints.

1. Introduction

In many applications, splines should not only pass through given data points (t_j, x_j), $j = 1, 2, \ldots, n$ but should also satisfy some additional conditions which are prescribed for the given problem. For example, if one wants to model a density function [cf. Feller (1971, p.3)] the model is useless unless it can produce non-negative values. A more general type of restriction is one which requires a fit to be "visually pleasing" [cf. Carlson (1987)] and the meaning of this may vary from problem to problem.

One approach to such a problem is motivated by physical arguments. The curve one is interested in is regarded as a flexible ruler or beam (draughtsman's spline) (cf. Malcolm (1977), de Boor (1978, p.67)) and thus, the strain energy has to be minimized, of course subject to the given restrictions.

The second approach is motivated more by practical arguments. In this approach one constructs a spline directly, such that it fits all the given requirements. This is ordinarily done by choosing additionally introduced parameters suitably.

For splines with no restrictions apart from the interpolatory conditions, the first approach was followed for example by Golomb and Jerome (1982). For non-negativity constraints in connection with the simplified strain energy $\int_a^b \ddot{x}(t)^2 dt$ see Opfer and Oberle (1988). For results in the case of minimizing $\int_a^b \{x^{(k)}(t)\}^2 dt$ restricted to $x^{(m)} \geq 0$ see Dontchev and Kalchev (1988). Recent contributions to the second approach are given by Rentrop and Wever (1988) and Schaback (1989). The case where the functional to be minimized and the constraints are all of general form is treated by Opfer (1989).

In this paper we shall follow the concept of minimizing the strain energy, which is expressed by the integral over the square of the curvature of the corresponding curve, subject to non-negativity constraints. Thus the curves to be investigated cannot penetrate the x-axis, and this type of problem may be regarded as an *obstacle* problem.

For brevity and simplicity our problem will be posed only for functions rather than curves.

2. The Problem

Given *data* (t_j, x_j), $j = 1, 2, \ldots, n \geq 2$, with $a = t_1 < t_2 < \cdots < t_n = b$ and n fixed, we require a function $x \in W_2^2$ with $x(t_j) = x_j$, $j = 1, 2, \ldots, n$, which, if parametrized with respect to arc length s, minimizes

$$f(x) = \int_0^{\ell} \kappa^2(s) ds \qquad (1)$$

subject to

$$x(s) \geq 0, \qquad (2)$$

where κ is the curvature of x and ℓ the length of the curve described by x. It should be pointed out that in this setting neither the length ℓ of the resulting curve x nor the interpolation conditions (which have the form $x(s_j) = x_j$) are known in advance, since the arc lengths s_j, which are the lengths from the first point to the point numbered j measured on x, are not known beforehand.

It is mentioned in various papers [e.g. Golomb and Jerome (1982)] that the corresponding Euler equation for the unrestricted case can be put (by suitable transformation) in the form of the pendulum equation, with the consequence that the minimum of the above mentioned functional (if it exists at all) is a piecewise elliptic integral. However, very few authors use this result. One exception is K.D. Reinsch (1981) who takes advantage of the existing procedures for elliptic functions presented by Bulirsch (1965).

Most of the numerical work in the unrestricted case was discussed by Malcolm (1977). In particular, results by Glass (1966), Larkin (1966), Lee and

Forsythe (1973), Mehlum (1969), Woodford (1969) are mentioned. We are not aware of any more recent published results.

The above setting in terms of the arc length s has the advantage of formal simplicity; the disadvantage already mentioned is that the upper bound for the integral in (1) and the interpolation conditions contain a quantity unknown a priori. In this paper we have decided to use rectangular coordinates and restrict our attention to curves which are graphs of functions.

Given $x \in C^2[a,b]$, the curvature of x at a point $t \in [a,b]$ is defined by

$$\kappa(x(t)) = \frac{\ddot{x}(t)}{(1+\dot{x}(t)^2)^{3/2}}. \tag{3}$$

If we use the transformation

$$ds = \sqrt{1+\dot{x}(t)^2}dt, \tag{4}$$

the above functional (1) reads

$$f(x) = \int_a^b \frac{\ddot{x}(t)^2 dt}{(1+\dot{x}(t)^2)^{5/2}}. \tag{5}$$

By the introduction of the "spaces"

$$\begin{aligned} M &= \{x \in W_2^2 : \quad x(t_j) = x_j,\ j = 1,2,\ldots,n\}, && (6)\\ H &= \{x \in W_2^2 : \quad x(t_j) = 0,\ j = 1,2,\ldots,n\}, && (7)\\ M^+ &= \{x \in M : \quad x \geq 0\}, && (8) \end{aligned}$$

we can give the problem the brief form:

$$\text{minimize } f(x) \text{ given by (5) subject to } x \in M^+. \tag{9}$$

For later reference we abbreviate the integrand of (5) as

$$F(x) = F(t,x,\dot{x},\ddot{x}) = \frac{\ddot{x}(t)^2}{(1+\dot{x}(t)^2)^{5/2}}. \tag{10}$$

There is one important subproblem which we term the *local* problem.

Local Problem. *Specify problem (9), but with $n = 2$ and with prescribed additional boundary conditions*

$$\dot{x}(t_1) = \dot{x}_1, \quad \dot{x}(t_2) = \dot{x}_2, \tag{11}$$

where $\dot{x}_1$, $\dot{x}_2$ are any given values.

If we could solve the local problem satisfactorily, we could already set up an algorithm for finding the solution of the general problem, which in this context we term the *global* problem. The algorithm would have the form:

Step 1: Solve the unrestricted problem, and call the solution x.

Step 2: Replace the solution x in those intervals $I_j = [t_j, t_{j+1}]$ in which x has negative values by the corresponding solution of the local problem, by using the computed derivatives at the endpoints as prescribed boundary conditions.

Step 3: Repeat Steps 1 and 2 by taking all derivatives at the given knots as unknowns, with the aim of producing a C^2-spline.

It should be noted that the first two steps alone already produce a C^1-spline. For the simplified strain energy $\int_a^b \ddot{x}(t)^2 dt$, this algorithm has been proposed by C. Reinsch (1988) and by Fischer, Opfer and Puri (1987). These authors show that the corresponding local solution can be computed explicitly in a very simple and efficient manner. For this simplified case, the algorithm has undergone extensive numerical testing by Dauner (1987) and Kröger (1989).

The local problem is a special case of the so-called *Hermite*-problem, in which, in addition to the already stated conditions, we require that $\dot{x}(t_j) = \dot{x}_j$, where $\dot{x}_j$, $j = 1, 2, \ldots, n$ are arbitrarily specified numbers.

3. Necessary Optimality Conditions

If one of the data points should fall below the x-axis, then our problem certainly has no solution. The following assumption on the data specification is therefore immediate.

Assumption 1. *Assume $M^+ \neq \emptyset$, i.e., $x_j \geq 0$ for all $j = 1, 2, \ldots, n$.*

For our first result we need two more "spaces", the definition of which depends only on $x \in M^+$, namely

$$H_x^+ = \{h \in H : \quad x + h \geq 0\}, \tag{12}$$

$$H_x^{++} = \{h \in H_x^+ : \quad x - h \geq 0\}. \tag{13}$$

The latter space H_x^{++} is usually referred to as the *envelope* or *hull* of x, since it contains all those h for which $|h| \leq x$. It has the properties

$$h \in H_x^{++} \iff \quad -h \in H_x^{++}, \tag{14a}$$

$$h \in H_x^{++}, x(\tau) = 0,\ a < \tau < b \quad \Longrightarrow \quad h(\tau) = \dot{h}(\tau) = 0 \leq \ddot{h}(\tau). \tag{14b}$$

Theorem 1. *Suppose that x_0 solves the problem stated in (9). Then*

$$f'(x_0, h) := \lim_{\substack{\alpha > 0 \\ \alpha \to 0}} \frac{1}{\alpha}\{f(x_0 + \alpha h) - f(x_0)\} = \int_a^b \{F_{\dot{x}}(x_0)\dot{h}(t) + F_{\ddot{x}}(x_0)\ddot{h}(t)\}dt$$

$$\geq 0 \ \textit{ for all } h \in H_{x_0}^+, \tag{a}$$

and

$$= 0 \ \textit{ for all } h \in H_{x_0}^{++}. \tag{b}$$

Proof: (a) By the definition of $H_{x_0}^{+}$ we have $x_0 + \alpha h \in M^+$ for all $0 \leq \alpha \leq 1$. Since in that case, by definition of x_0, we have $f(x_0) \leq f(x_0 + \alpha h)$, the assertion follows from the definition of the derivative of f.
(b) This follows from (a), by using (14a) in connection with the linearity of $f'(x_0, \cdot)$ with respect to h. ■

In order to proceed from Theorem 1 we choose a fixed subinterval $I_j = [t_j, t_{j+1}]$, $j = 1, 2, \ldots, n-1$ and an $h \in H_{x_0}^{++}$ which vanishes outside I_j. If $j > 1$ and $j + 1 < n$ then $h \in W_2^2$ implies $h^{(\ell)}(t_j) = h^{(\ell)}(t_{j+1}) = 0$ for $\ell = 0, 1$. By applying partial integration, Theorem 1 implies that for all $j = 1, 2, \ldots, n-1$

$$\int_{t_j}^{t_j+1} \tilde{F}(x_0)\ddot{h}(t)dt = 0, \text{ for } h \in H_{x_0}^{++}, \text{ and } h(t) = 0 \text{ for } t \notin I_j, \tag{15}$$

where in each interval

$$\tilde{F}(x_0) = F_{\ddot{x}}(x_0) - v, \tag{16a}$$

$$\dot{v} = F_{\dot{x}}(x_0), \tag{16b}$$

and

$$v(a) = v(b) = 0, \tag{16c}$$

where F is defined in (10). In order to obtain (15) we use the relation

$$\int_{t_j}^{t_j+1} F_{\dot{x}}(x_0)\dot{h}(t) = \dot{h}(t)v(t)\Big|_{t_j}^{t_j+1} - \int_{t_j}^{t_j+1} v(t)\ddot{h}(t)dt$$
$$= -\int_{t_j}^{t_j+1} v(t)\ddot{h}(t)dt.$$

If $1 < j < n-1$ then $\dot{h}(t_j) = 0$, and thus the above constant parts vanish. If, however, $j = 1$ or $j = n$ then $\dot{h}(t_j)$ may take any value. In this case we note the fact that v defined in (16b) is determined only up to a constant, and this allows us to set $v(a) = v(b) = 0$, at least provided that we have $n > 2$.

For later use we compute the partial derivatives of F:

$$F_{\dot{x}} = \frac{-5\dot{x}\ddot{x}^2}{(1+\dot{x}^2)^{7/2}}, \quad F_{\ddot{x}} = \frac{2\ddot{x}}{(1+\dot{x}^2)^{5/2}}. \tag{17}$$

Theorem 2. *(a) For $n = 2$ the solution x_0 of (9) is the linear interpolant of the two data (t_1, x_1) and (t_2, x_2). (b) For $n > 2$, if x_0 solves the problem (9) and if v is defined as in (16b) and (16c), then*

(i) $F_{\ddot{x}}(a) = F_{\ddot{x}}(b) = 0$,
(ii) $F_{\ddot{x}} - v$ *is linear in* $I_j = [t_j, t_{j+1}]$, *provided* $x_0 > 0$ *in* I_j,
(iii) $F_{\ddot{x}} - v$ *is linear between interior zeros of* x_0, *and linear between interior zeros and knots, where an interior zero of* x_0 *is a zero in the open interval* $]t_j, t_{j+1}[$,
(iv) $\frac{d}{dt}(F_{\ddot{x}} - v)$ *always has non-negative jumps at the interior zeros of* x_0.

Proof: (a) In this case the proof follows because $x_0 \in M^+$ and $f(x_0) = 0$. (b) An application of a theorem of du Bois-Reymond tells us that in each subinterval I_j the expression $\tilde{F}$ is a polynomial of degree 1 in t. In addition, if τ is a zero of x_0 then τ has to be introduced as a *new knot*, since we have $h \in H^{++}_{x_0}$ together with the derivative $\dot{h}$ vanishing at τ. The remaining details of the proof follow the corresponding proofs in Opfer and Oberle (1988). ■

Condition (i) corresponds to the so-called "natural boundary conditions" of an ordinary spline. In the case of an ordinary cubic spline, condition (ii) is the condition that the second derivative should be piecewise linear. The remaining two conditions cover the constraints. Condition (iii) implies that any zero of the solution x_0 must be considered as a new knot. The jump condition (iv) means that the difference between the right and left derivative at a new knot must always be non-negative.

Condition (ii) reads explicitly

$$2\ddot{x}(t)/(1+\dot{x}(t)^2)^{5/2} - v(t) = At + B, \qquad (18)$$

where

$$-5\dot{x}\ddot{x}^2/(1+\dot{x}^2)^{7/2} = \dot{v}.$$

Differentiation with respect to t yields

$$(2(1+\dot{x}^2)x^{(iii)} - 5\dot{x}\ddot{x}^2)/(1+\dot{x}^2)^{7/2} = A. \qquad (19)$$

If we differentiate once more we obtain Euler's equation

$$(1+\dot{x}^2)\{2(1+\dot{x}^2)x^{(iv)} - 20\dot{x}\ddot{x}x^{(iii)} - 5\ddot{x}^3\} + 35\dot{x}^2\ddot{x}^3 = 0. \qquad (20)$$

If $\ddot{x} \neq 0$ we may multiply equation (19) by $\ddot{x}$ and obtain

$$\frac{d}{dt}\frac{\ddot{x}^2}{(1+\dot{x}^2)^{5/2}} = \frac{2\ddot{x}x^{(iii)}}{(1+\dot{x}^2)^{5/2}} - \frac{5\dot{x}\ddot{x}^3}{(1+\dot{x}^2)^{7/2}} = A\ddot{x}.$$

Thus, by integrating we obtain

$$(A\dot{x} + C)(1+\dot{x}^2)^{5/2} = \ddot{x}^2, \qquad (21)$$

which is Woodford's (1969, eq. (5)) equation.

The first three differential equations, valid in each subinterval and connected by suitable boundary values can be written as first order systems which are more convenient for computing. Equation (18) reads

$$\dot{x} = u, \qquad (18a)$$

$$\dot{u} = 0.5(1+u^2)^{5/2}(v + At + B), \qquad (18b)$$

$$\dot{v} = -1.25u(1+u^2)^{3/2}(v + At + B)^2. \qquad (18c)$$

Equation (19) takes the form

$$\dot{x} = u, \qquad (19a)$$

$$\dot{u} = v, \qquad (19b)$$

$$\dot{v} = 0.5A(1+u^2)^{5/2} + 2.5uv^2/(1+u^2). \tag{19c}$$

Finally, Euler's equation (20) is equivalent to

$$\dot{x} = u, \tag{20a}$$

$$\dot{u} = v, \tag{20b}$$

$$\dot{v} = w, \tag{20c}$$

$$\dot{w} = (10uvw + 2.5v^3)/(1+u^2) - 17.5u^2v^3/(1+u^2)^2. \tag{20d}$$

In addition to these equations we have the boundary conditions (16c), part *(i)* of Theorem 2, the interpolation conditions and the smoothness conditions at the interior knots. Moreover we have to introduce new knots τ, $t_j < \tau < t_{j+1}$ (for some j), whenever τ is an interior zero of x_0. For such points τ, we have the additional conditions $\dot{x}_0(\tau) = 0$, $\ddot{x}_0(\tau) \geq 0$. The latter conditions reflect the fact that x_0 has not only a zero at τ but also a minimum.

The implied algorithm requires the repeated solution of one of the above sets of differential equations with additional known and unknown boundary conditions. Results in this direction will be published elsewhere.

Acknowledgments

The author would like to thank the editors for valuable help in preparing this manuscript.

References

1. de Boor, C., *A Practical Guide to Splines*, Springer, New York, Heidelberg, Berlin, 1978.
2. Bulirsch, R., Numerical Calculation of Elliptic Integrals and Elliptic Functions, Numer. Math. 7, 1965, 78–90.
3. Carlson, R.E., Shape Preserving Interpolation, in: Mason, J.C. and Cox, M.G. (eds.): *Algorithms for Approximation*, Proceedings of IMA Conference on Algorithms for the Approximation of Functions and Data, Shrivenham, England (1985), Clarendon Press, Oxford, 1987, 97–114.
4. Dauner, H., Formerhaltende Interpolation mit kubischen Splinefunktionen, Diploma Thesis, Technische Universität München, 1987.
5. Dontchev, A.L. and Kalchev, Bl., On Convex and Nonnegative Best Interpolation, C. R. Acad. Bulgare Sci. 41 (1988), 21–24.
6. Feller, W., *An Introduction to Probability Theory and its Applications*, Vol. 2, 2nd ed., Wiley, New York, London, Sydney, Toronto, 1971.
7. Fischer, B., Opfer, G. and Puri, M.L., A Local Algorithm for Constructing Non-negative Cubic Splines, Hamburger Beiträge zur Angewandten Mathematik, Reihe A, Preprint 7, 1987, 18p, to appear in J. Approx. Theory.
8. Glass, J.M., Smooth-Curve Interpolation: A Generalized Spline-Fit Procedure, BIT 6 (1966), 277–293.

9. Golomb, M. and Jerome, J., Equilibria of the Curvature Functional and Manifolds of Nonlinear Interpolating Spline Curves, SIAM J. Math. Anal. **13** (1982), 421–458.
10. Kröger, N., Nicht-negative Splines und deren numerische Berechnung, Diploma Thesis, Universität Hamburg, 1989.
11. Larkin, F.M., An Interpolation Procedure Based on Fitting Elastica to Given Data Points, COS Note 5/66, Theory Division, Culham Laboratory, Abingdon, England, 1966, 15 p.
12. Lee, E.H. and Forsythe, G.E., Variational Study of Nonlinear Spline Curves, SIAM Review **15** (1973), 120–133.
13. Malcolm, M.A., On the Computation of Nonlinear Spline Functions, SIAM J. Numer. Anal. **14** (1977), 254–282.
14. Mehlum, E., Curve and Surface Fitting Based on Variational Criteriae for Smoothness, Central Institute for Industrial Research, Oslo, 1969, 83 p.
15. Opfer, G., Necessary Optimality Conditions for Splines under Constraints, in: Chui, C.K., Schumaker, L.L., Ward, J.D. (eds.): *Approximation Theory VI*, Academic Press, New York, 1989, to appear.
16. Opfer, G. and Oberle, H.J., The Derivation of Cubic Splines with Obstacles by Methods of Optimization and Optimal Control, Numer. Math. **52** (1988), 17–31.
17. Reinsch, C., Software for Shape Preserving Spline Interpolation, Mscr. 1988, 12p.
18. Reinsch, K.D., Numerische Berechnung von Biegelinien in der Ebene, Dissertation, Technische Universität München, 1981, 246 p.
19. Rentrop, P. and Wever, U., Computational Strategies for the Tension Parameters of the Exponential Spline, in: Bulirsch, R., Miele, A., Stoer, J., Well, K. (eds.):*Optimal Contol*, Lecture Notes in Control and Information Sciences, **95** (1987), Springer, Berlin, Heidelberg, New York, London, Paris, Tokyo, 122–134.
20. Schaback, R., Convergence of Planar Curve Interpolation Schemes, in: Chui, C.K., Schumaker, L.L., Ward, J.D. (eds.): *Approximation Theory VI*, Academic Press, New York, 1989, to appear.
21. Woodford, C.H., Smooth Curve Interpolation, BIT **9**, (1969), 69–77.

SPLINE CURVE FITTING OF DIGITIZED CONTOURS

C. POTIER and C. VERCKEN
Département Informatique
Ecole Nationale Supérieure des Télécommunications

Abstract This paper presents an algorithm for finding a mathematical curve approximation to a digitized contour. The whole application consists in processing optically scanned graphic documents to store them in a data base for consultation and manipulation. For each contour, an approximating parametric cubic spline curve, with as few control points as possible, is determined by minimizing a smoothing criterion. The algorithm is particularly efficient for large data sets, the number of control points always being very small. To handle the curve, the B-spline representation is associated to a hierarchical data structure, obtained by generating the Bézier points and subdividing the corresponding Bézier segments.

Key words : Cubic splines, Curve fitting, Data compression.

1. Introduction

The algorithm presented is a part of a software project at our school concerned with the consultation and manipulation of documents archived in digital form. Documents which were not created in this form are to be converted to a compatible form. To digitize them, we use a scanner whose resolution is 300 dots per inch. The output, a binary image, is an array of pixels (approximately 2400*3500 for a A4 document) without any information about the structure of the document. For archiving purposes, data compression is necessary, and even run length encoded forms such as CCITT group III or IV, used for facsimile coding and transmitting, represent a very large amount of data and give no information about the structure of the document. For black and white graphics an analytical representation of contours is useful for data compression, shape description and geometrical transformation. To deal with complex documents it would be necessary to have preliminary processing, which is outside the scope of this paper, such as segmentation into text, black and white graphics and half tone zones as described by Ito and Saakatani (1982) and determination of the connected components of the graphic zones as described by Ronse and Devijver (1984). Since we presently have studied only very simple graphics including a single component, a minimal basic preprocessing is necessary to determine an ordered list of black pixels representing a contour or a curve and to convert it into a continuous boundary by vectorization. The next step consists in finding a good spline approximation by first determining a small accurate set of knots and then adjusting the B-spline coefficients.

2. Contour extraction and polygonal approximation

Let us recall some basics in binary image processing which are given, for instance, by Pavlidis (1982) :

- Background pixels (white) have value 0 and object pixels (black) have value 1.
- An "object" connected component is a set P of black pixels adjacent to an edge (D-neighbours) or to a corner (I-neighbours) .

A connected component is entirely determined by its external contour and possibly by internal ones if there are "holes".

2.1 Contour extraction

Definition : Let P be a connected component. The contours C of P are the sets of adjacent pixels in P which have at least one D-neighbor not in P.

A contour C can be represented by a closed path that can be determined by the following algorithm proposed by Pavlidis (1982) : "An observer walks along pixels belonging to the set and selects the rightmost available until the current pixel is the initial pixel". This initial pixel is usually found by a top-to-bottom, left-to-right scan. The algorithm must be applied for each contour of the set P.

2.2 Line thinning

Given the scanner resolution and the graphics attributes (pen-size) a line is scarcely ever one pixel wide and often has interior pixels. Since the contour is adequate to describe a filled shape but not a line, it is necessary to distinguish shapes from lines that are described by the skeleton of the component and its width. A maximum thickness T is set and the contour is peeled off by removing all contours points that are not "skeletal", following Pavlidis (1982). This connectivity-preserving peeling is repeated at most T times and, if during this process no points can be removed, the skeleton is obtained and the component is considered as a line of thickness $T' \leq T$; otherwise it is a filled shape. Henceforth, line skeletons and shape contours will be processed as closed or open lines.

2.3 Vectorization

The next step consists in converting the discrete pixel-chain representation of lines, possibly closed, into a continuous very fine polygonal approximation (vectorization) which vertices are the end points of "straight line" segments of the grid. We determine the vertices by using the sequential method, based on the minimal area deviation criterion, proposed by Wall and Danielsson (1984), with a small tolerance value S :

Algorithm 1:

```
(a) Let $M_0(x_0,y_0)$ be the starting point and $M_1$ its contour successor.
Initiate the algebraic area $f_1$ of $(M_0M_0M_1)$ to 0. Let $i = 2$.
(b) Calculate the algebraic area $\Delta f_i$ of $(M_0M_{i-1}M_i)$ and the cumulative area $f_i=f_{i-1}+\Delta f_i$.
Calculate the length $L_i$ from $M_0$ to $M_i$.
(c) if $f_i \leq S\ L_i$ then increment i and repeat (b).
Otherwise put $M_{i-1}$ in the set of vertices and take it as new starting point.
```

The detection of "characteristic points" is done simultaneously. At any time we keep the segment's general direction, as one of the 8-connectivity, and we mark the points at which a true direction change (not temporary for a single point) is found. These "characteristic points" are taken as new starting points rather than the last point satisfying the criterion .

3. Curve fitting

We want to determine a parametric curve, fitted to the ordered sampled points $\{P_i\}_{i=1}^{N}$ obtained by the preprocessing, which can be written using B-spline basis: $M(t) = \sum_{i=1}^{n} M_i\ B_{i,4}(t)$, for $t \in [T_0, T_1]$, with the number n of basis functions as small as possible and where $B_{i,4}$ are cubic normalized B-splines .

To fit a parametric curve to a set of points $\{P_i\}_{i=1}^{N}$, any data point P_i must be assigned a parameter value u_i $(u_1<u_2<\ldots<u_N)$ that we suppose to vary from 0 to T for sake of simplicity. The chord-length parametrization needs more arithmetic operations than

uniform parametrization but it is more natural since the points P_i are unevenly spaced and the contour of the curve is very close to the polyline joining the points P_i.

3.1 Knots determination

To determine n basis functions $\{B_{i,4}\}_{i=1}^{n}$, a subdivision $t_1 \leq t_2 \leq \ldots \leq t_{n+4}$ of $[0,T]$ is necessary and is automatically searched while keeping a trade-off between the closeness of the fit and the number of basis functions. There must be as few knots as possible but the mean square distance to the data points must be no greater than a tolerance level. Three ways to obtain the knots placement have been compared:

- In the first method, the subdivision $\{t_j\}$ is determined by using the minimal area deviation criterion of Wall and Danielsson (1984), as described above, with a sufficiently large tolerance value S.

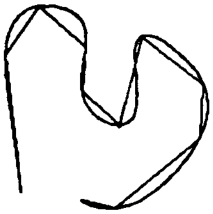


Figure 1: Subdivision obtained by algorithm 1

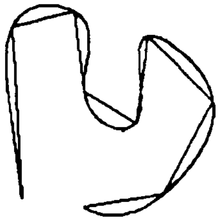


Figure 2: Subdivision obtained by algorithm 1 used twice

The subdivision $\{t_j\}_{j=1}^{n}$ obtained by this method is not symmetric, the "beginning" and the "end" of the sampling do not play the same part (fig.1). To avoid this, it is possible to use the algorithm twice, forward and back, and to calculate the mean of the two subdivisions. The subdivision obtained by using this algorithm twice is more regular (fig.2).

- In the second method, knots are added until, on each subinterval $I=[T_i,T_{i+1}]$, E_I the mean square distance of points $P_j \in I$ to the least-squares straight line is less than a parameter value E_{max}.

Algorithm 2

(a) Start with one subinterval $[T_1,T_2]$; call M the collection of subintervals.

(b) On each $I \in M$, calculate the rms E_I to the least-squares straight line.

(c) If $E_I > E_{max}$ then split I in two and put both pieces into M, otherwise increment I. The algorithm is repeated from (b) until all subintervals are processed.

The subdivision obtained with this algorithm (fig.3) is not very regular, some intervals seem to be too small. This algorithm should be improved with a "merge" algorithm (see

Pavlidis,1974) to merge two consecutive intervals if possible. Split and merge algorithms produce good results but are time-consuming since they require multiple passes through the data.

- The third method is based on least-squares fitting by cubic polynomials in X and Y.

Algorihm 3

Let S be a parameter value.

(a) Start with one subinterval $[T_1,T_2]$; call M the collection of subintervals.

(b) On each interval $I \in M$, calculate 2 least-squares cubic polynomials (P_x and P_y) for X and Y.

Calculate $D_x=\Sigma_j (P_x (u_j) - X_j)^2$ and $D_y=\Sigma_j (P_y (u_j) - Y_j)^2$.

(c) If $D_x>S$ or $D_y>S$ then split the interval in two and put both pieces into M, otherwise increment I. Return to (b).

(d) Subdivide each interval $[T_i,T_{i+1}]$ into 4 to obtain the subdivision $t_1 \le t_2 \le \ldots \le t_{n+4}$.

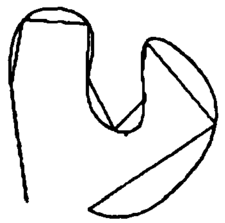


Figure 3 : Subdivision obtained by algorithm 2

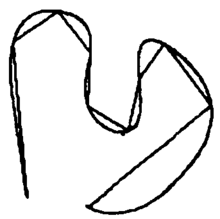


Figure 4 : Subdivision obtained by algorithm 3

We use one of these algorithms to obtain a subdivision of intervals between two "characteristic" points: either endpoints or points marked during the preprocessing. After obtaining the subdivision, we add multiple (triple) knots corresponding to marked points. Multiple knots are also added at the ends of an open curve and cyclic knots are added if the curve is closed.

In all cases, the number of knots n+4 is much smaller than the number N of data points P_i and generally subdivisions are quite similar (fig.1 to 4). Nevertheless algorithm 3 should produce better results in case of lightly snaky data. With algorithm 1 the subdivision is not optimal; however this method is attractive since it is less time-consuming as shown in Tab.1.

Table 1: Time to obtain the subdivision.

Algorithm	1	2	3
Time	3 seconds	6 seconds	25 seconds

3.2 Fitting a parametric spline curve

The third step of the fitting process is to determine the parametric spline which must be close to the points P_i and "smooth". The control points $\{M_i\}_{i=1}^{n}$ are searched while minimizing the quadratic functional:

$$I_2(M) = \sum_{j=1}^{N} [M(u_j) - P_j]^2 + \mu \int_0^T \{ [X'']^2 + [Y'']^2 \} \, dt$$

The positive parameter μ controls the trade-off between the sum of the distances of points $\{P_i\}_{i=1}^{N}$ from the curve, measured by the first term of I_2, and the smoothness of the curve, measured by the integral, which is a rough approximation to the curvature. However this functional I_2 involves two independent minimization problems to determine X and Y. The solution is easily obtained (Potier and Vercken, 1985) by solving two linear systems : $Qa=z_x$ and $Qb=z_y$ where the matrix Q is a symmetric positive matrix. Moreover, in the case of open curves, Q is banded, whereas in the case of closed curves there are non-zero terms at opposite corners, as shown in figure 5 (in case of quadratic splines). The solution of both systems can be obtained by factorizing the matrix Q with a data structure adaptated to the matrix (fig.6) and takes O(n) operations.

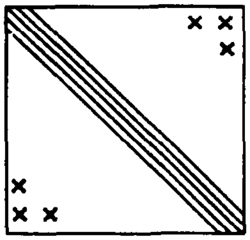


Figure 5 : Matrix Q in cyclic case

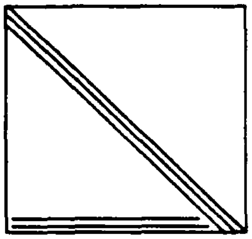


Figure 6 : Cholesky factorisation

The parameter μ is adjusted so that the mean square distance to the data points is no greater than a value T which may be a number of pixels. In figures 7 and 8 the value of T corresponds approximately to a tolerance of 2 pixels. On both figures, the initial curve, the B-spline approximation and the control polygon are drawn. The initial curve of figure 7 was designed with 9 B-splines, the digitized picture contained about 5000 pixels and the skeleton 1139 pixels, the vectorization gave 120 segments. The fitting by 11 B-splines, with knots obtained from algorithm 1, is very accurate.

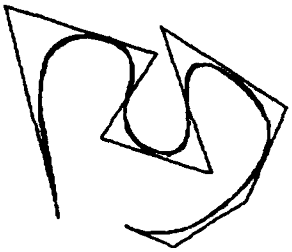


Figure 7 : 1139 pixels
and 11 B-splines

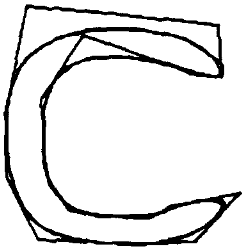


Figure 8 : 781 pixels
and 20 B-splines

However we did not succeed in finding a good approximation with fewer basis functions. However, figure 8 designed with 16 B-splines, algorithm 3 gave better subdivision than algorithm 1. The fitting with 16 B-splines kept the initial shape but smoothed the corners.

4. Interactive Curve Handling

This smoothing algorithm was integrated in an interactive graphics environment where performance of the algorithms is of fundamental importance. To make geometrical transformations on a displayed curve, the user has to identify the curve on the screen with a cursor pointed "near" the curve. The selection is done by testing the convex hull of the cubic Bézier composite representation of the curve obtained by the algorithm of Böhm (1981). By applying the De Casteljau algorithm, each cubic Bézier segment is split in two, if necessary, and this process is recursively applied. Successive splittings involve a hierarchical data representation of the curve (fig.9).

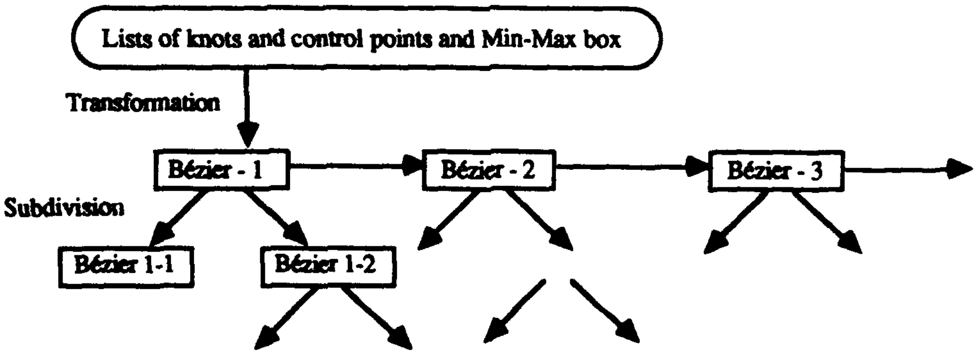


Figure 9 : Hierarchical data representation

The cubic Bézier representation of the curve can be used for printing "nicely" the curve on a PostScript-laserprinter since the PostScript operator "curveto" adds a cubic Bézier section to the current curve.

5. Conclusion

The fitting algorithm presented in this paper is very efficient and can easily be inserted in graphic documents processing software. The knots determination, which is crucial, will be developed and the knot removal algorithm of Lyche and Morken (1987) will be compared with other methods.

Moreover it would be very useful to extend this algorithm to more basis functions such as non-uniform rational B-splines to fit straight lines and conics.

References

Böhm W., Generating the Bézier points of B-spline curves and surfaces, Computer Aided Design, 13, n°6, November 1981, 365-366.

Ito S. and Saakatani S., Field Segmentation in Document Image, 6th IJCPR, Munich, 1982, 492-495.

Lyche T. and Morken K., Knot Removal for Parametric B-splines Curves and Surfaces,CAGD 4, 1987, 217-230.

Pavlidis T. and Horowitz S.T., Segmentation of Plane Curves, IEEE Transactions on Computers, 23, n°8, August 1974, 860-870.

Pavlidis T., Algorithm for Graphics and Image Processing, Washington D.C., Computer Science Press, 1982.

C. Potier and C. Vercken, Lissage de Surfaces par Eléments Finis, L'Echo des Recherches, n° 122, 1985, p. 51-58.

Ronse C. and Devijver P.A., Connected Components in Binary Images: The detection Problem, Research Studies Press,1984.

Wall K. and Danielsson P.E., A Fast Sequential Method for Polygonal Approximation of Digitized Curves, Graphics and Image Processing, 28, 1984, 220-227.

A B-SPLINE APPROXIMATION ALGORITHM FOR QUASI-INTERPOLATION OR FILTERING.

CHRISTOPHE RABUT
Institut National des Sciences Appliquées
Toulouse, France.

Abstract We propose some straightforward modifications to the usual B-spline algorithm for the approximation of discrete data. The modifications permit quasi-interpolation or filtering of the data. We propose a simple algorithm for the computation of "good" nodes - a linear combination of neighbourhood data, with suitable coefficients - in order to obtain desirable properties for the global approximation process. To analyse this process - and to determine the coefficients of the linear combination - we use the transfer function of the associated filter, which is a very good tool to obtain a global view of the process (i.e., which frequencies of the data are amplified, or attenuated, and how much, etc.).

Key words B-spline approximation, Quasi-interpolation, Filtering, Transfer function

1. Introduction

Given discrete equidistant data (x_i, y_i), it is a common requirement to determine the B-spline approximation of those data, i.e. to perform the following algorithm:

```
FOR wanted x,
    compute $\sigma(x) = \sum_i y_i B_i(x)$
```

where the B_i are the usual cubic B-splines, centered at x_i; in this paper we will consider only cubic splines, but the ideas may be extended to any degree.

The important advantage of the above computation is that the curve so-obtained is very quickly determined, and follows the shape of the data (precisely, we have $\sigma(x_j) = (y_{j+1} + 4y_j + y_{j-1})/6$. But there is also an important drawback: such an algorithm allows no flexibility, in the sense that we cannot force the curve to be nearer to the data, or, conversely, we cannot use it to filter noise which may be present in the data.

To overcome this drawback, we propose splitting up the nodes, Y_i^n , of the B-spline approximation, from the data y_i ; for fast computation, we will require Y_i^n to be a linear combination of 2n+1 of the y_j placed around y_i (the coefficients of the linear combination, as well as the value of n, being chosen according to the type of result to be obtained, see later). So, we propose performing the following algorithm:

```
For concerned i
    compute $Y_i^n = \sum_{j=-n}^{n} \alpha_j^n y_{i-j}$.
For wanted x
    compute $\sigma^n(x) = \sum_{i \in \mathbb{Z}} Y_i^n B_i(x)$.
```

The aim of this paper is to show some way of choosing appropriate coefficients of the linear combination, and, in order to do so, to present and use the "transfer function" (which is also called "attenuation function" in some papers), which is a very convenient and comprehensive tool to obtain a global view of the approximation process.

To do so, we will suppose we have an infinite regularly spaced data set (it is easy, if necessary, to extend a finite set to an infinite one, in an appropriate way), so $\forall\ i \in \mathbb{Z}$, $x_i = ih$. Also, we will take symmetric combinations ($\forall\ j \in [1, n]$, $a^n_{-j} = a^n_j$), so we will use the equivalent form $Y_i^n = \sum_{j=0}^{n} b_j (\delta^{2j} y)_i$, where

$$(\delta^2 y)_i = y_{i+1} - 2y_i + y_{i-1}.$$

2. Tranfer function

2.1 Definition

Presentation The first idea is to study only the result of the approximation scheme at the data: we will look only at $(\sigma^n(x_i))_{i \in \mathbb{Z}}$, and not at $\sigma^n(x)$ for $x \neq x_i$. In other words, we will study the transformation T_n: $y \in \mathbb{R}^{\mathbb{Z}} \longrightarrow z \in \mathbb{R}^{\mathbb{Z}}$ defined by

$$\forall\, j \in \mathbb{Z},\ z_j = \sigma^n(x_j) = \sum_{i\in\mathbb{Z}} \left(\sum_{k=-n}^{n} b_k^n \delta^{2k} y \right)_i B(x_{j-i}).$$

Since the values $\sigma(x_i)$ provide information on $\sigma(x)$, studying this discrete transformation will provide knowledge of the whole approximation process.

Obviously T_n is a linear transformation, and the vector $y = e_\alpha = (e^{2i\pi\alpha k})_{k\in\mathbb{Z}}$ (here $i^2 = -1$) is an eigenvector of T_n.

Definition Let us call $H_n(\alpha)$ the eigenvalue of the above transformation T_n, associated with the eigenvector e_α. The application $H_n : \alpha \longrightarrow H_n(\alpha)$ is called the "transfer function" of the filter T_n.

H_n is a real-valued function which is even and periodic (of period $1/h$).

Remarks: By definition, $H_n(\alpha)$ is the coefficient of amplification of a (co)sinusoïdal signal of frequency α. $H_n(0)$ is the coefficient of amplification of constants (we will always require $H_n(0)=1$!).

As $(e_\alpha)_{\alpha \in \mathbb{R}}$ are the only eigenvectors of T_n, it is understandable that the set of eigenvalues $(H_n(\alpha))_{\alpha \in \mathbb{R}}$ - i.e. the function H_n - is indicative of the whole transformation T_n.
As a particular case, if $\forall\, \alpha \in \mathbb{R}$, $H_n(\alpha) = 1$, then T_n is the identity.

2.2. Use of transfer function

Evaluation of H_n: theorem

Let H_n be defined as above.

Let $b_{n+1}^n = 0$. Then

$$H_n(\alpha) = b_0^n + \sum_{k=1}^{n+1} (b_k^n + b_{k-1}^n/6)(-4\sin^2 \pi\alpha h)^k;$$

The proof is quite easy: since $\forall\, y \in \mathbb{R}^{\mathbb{Z}}$,

$T_n(y) = b_0^n y + \sum_{k=1}^{n+1} (b_k^n + b_{k-1}^n/6)\delta^{2k} y$; so if g is the elementary filter $y \longrightarrow \delta^{2k} y$ (whose transfer function is $G(\alpha) = -4\sin^2 \pi\alpha h$),
$T_n = b_0^n g^0 + \sum_{k=1}^{n+1} (b_k^n + b_{k-1}^n/6) g^k$, and so $H_n = b_0^n + \sum_{k=1}^{n+1} (b_k^n + b_{k-1}^n/6) G^k$.

Error majoration: theorem

Let g and h be two filters, with respective transfer functions G and H.

Let $y \xrightarrow{g} z$ and $y \xrightarrow{h} z'$.

For any periodic function f of period P, let $\|f\|_2 = \sqrt{\int_P (f(x))^2 dx}$.

$$\text{If } y \in \ell^2, \quad \|z-z'\|_{\ell^\infty} \leqslant \|z-z'\|_{\ell^2} \leqslant \|G-H\|_\infty . \|y\|_{\ell^2} .$$

$$\text{If } y \in \ell^1, \quad \|z-z'\|_{\ell^\infty} \leqslant \|z-z'\|_{\ell^2} \leqslant \|G-H\|_2 . \|y\|_{\ell^1} .$$

3. Choosing the coefficients

Of course, the actual values of b_j^n will determine the properties of T_n, and so of the approximation process. The aim of this section is to present some criteria in order to determine interesting values of b_j^n. To do so, having chosen a transfer function G, we want to determine b_j^n so that the associated function H_n is as close to G as possible (if we want to quasi-interpolate data, we choose $G \equiv 1$, if we want to filter data as much as possible, we will choose $G(0) = 1$, $G(\alpha) = 0$ for $\alpha \neq 0$).

3.1. First criterion: truncated development

Definition The b_j^n are chosen in order that the first terms of the limited development at some point α_0 (for example $\alpha_0 = 0$) of H_n coincide with those of G; so, if G is written as

$G(\alpha) = \sum_{k \in \mathbb{N}} c_k (-4 \sin^2 \pi\alpha h)^k$, we obtain ($\alpha_0 = 0$):

$$\begin{cases} \quad b_0^n = c_0 \\ j = 1,\dots,n : b_j^n = c_j - b_{j-1}^n/6 \\ j \geqslant n+1 \quad : b_j = 0 \end{cases}$$

For quasi-interpolating data ($G \equiv 1$), we obtain

$$\forall k \in [0,n], \ b_k^n = (-1/6)^k$$

For a high-cut filter ($G(0) = 1$, $\forall k \in \mathbb{N}$, $G^{(k)}(1/2h) = 0$), we obtain:

$$\forall k \in [0,n] \ , \ b_k^n = \binom{n}{k} (1/4)^k \qquad (\Longleftrightarrow \forall k \in [0,n] \ , \ a_k^n = \binom{2n}{n+k} (1/4)^k).$$

Error majoration: theorem

Let $n \in \mathbb{N}$

Let $\forall j \in [0,n]$, $b_j^n = (-1/6)^j$; $\forall j \in [0,n]$, $b_j^n = 0$.

Let $\sigma^n(x) = \sum_{i \in \mathbb{Z}} \left(\sum_{k=0}^{n} b_k^n \delta^{2k} y \right)_i B(x - x_i)$.

Then $\forall j \in \mathbb{Z},\quad \sigma^n(x_j) = y_j - (-1/6)^{n+1}(\delta^{2n+2}y)_j$.

As a consequence majorations of $|\sigma^n(x_j) - y_j|$ may easily be obtained, and, in particular, σ^n interpolates any polynomial of degree at most 2n+1.

<u>Proof</u>: induction on n.

Figures: Every figure is provided for n = 0, 1, 2, 3, 4 and 5. Figures 1 and 2 are transfer functions of a quasi-interpolation and of a high-cut filter.
Figures 3 and 4 are σ^n for some data: (the □ are the data; the + are the computed nodes Y_i^n). Figure 3 is for step data: $i \leq 0 \implies y_i = 0$, $i \geq 1 \implies y_i = 1$. Figure 4 is for a parametric curve: each component is calculated separately (equidistant parameter).

3.2. Second criterion: least squares

The truncated development criterion is very interesting as the formulae obtained are very simple and lead to a many interesting theorical and practical results. But other criteria give better results in most cases. An example of such a criterion is given in this section.

3.2.1. Criterion

Looking at theorem 2.3, the idea now is to minimize $\|H_n - G\|_2$. This is quite simple as $E_n = \int_0^{1/2h}(G(\alpha) - H_n(\alpha))^2 d\alpha$ is a quadratic function of the coefficients b_j^n. Minimising E_n leads to the solution of a linear system with unknowns b_j^n.

Of course, we can add some additional constraints, such as, for example:

- reproducing constants: $H_n(0) = 1$ $\quad (\Longleftrightarrow b_0^n = 1,\ b_1^n = -1/6)$
- reproducing polynomials of degree ≤ 3: $H_n(0) = 1$; $H''_n(0) = 0$ $\quad (\Longleftrightarrow b_0^n = 1\ ,\ b_1^n = -1/6\ )$
- no frequency amplified: $\forall\alpha \in \mathbb{R}\ ,\ H_n(\alpha) \leq 1$
- no frequency inversed: $\forall\alpha \in \mathbb{R}\ ,\ H_n(\alpha) \geq 0$.

Here are some values obtained for quasi interpolation (i.e. $G(\alpha) \equiv 1$), with the reproducing constants constraint.

n=1		$b_0^1=1$	$b_1^1=-8/23$			
n=2	d=474	$b_0^2=1$	$b_1^2=-30/d$	$b_2^2=47/d$		
n=3	d=8838	$b_0^3=1$	$b_1^3=-1890/d$	$b_2^3=-351/d$	$b_3^3=-240/d$	
n=4	d=15443	$b_0^4=1$	$b_1^4=-22776/d$	$b_2^4=11320/d$	$b_3^4=4002/d$	$b_4^4=1137/d$

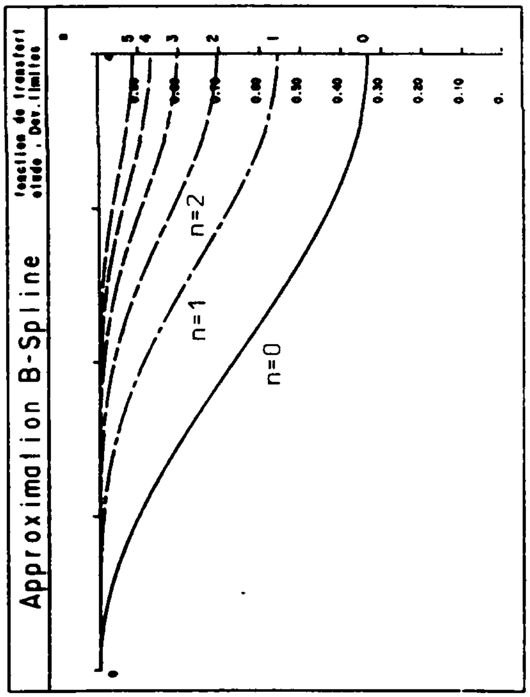


Figure 1

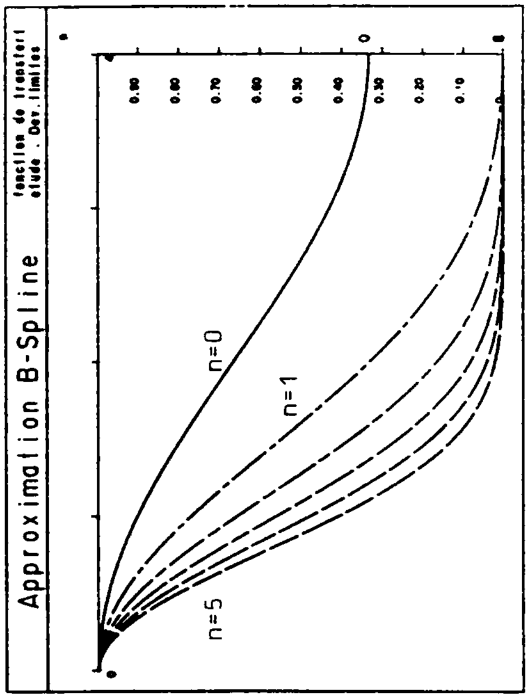


Figure 2

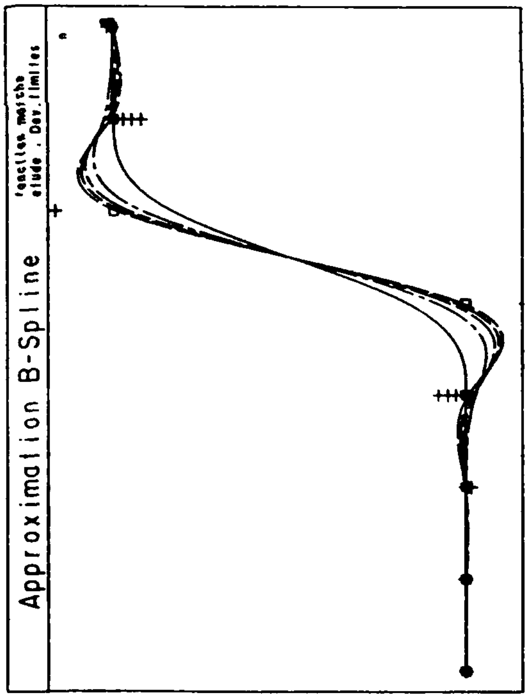


Figure 3

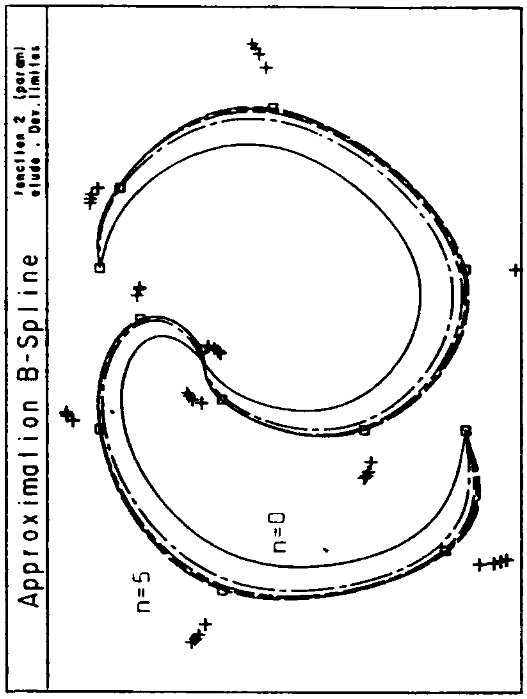


Figure 4

Figures 1 to 4 : Truncated developpement criterion.

We notice that, for $n = 1$, with $\alpha_0 = (1/\pi h)\sin^{-1}(5/8) \simeq (1/h)0.215$, we have $\underset{\alpha\in\mathbb{R}}{\mathrm{Max}}(H_1(\alpha)) = H_1(\alpha_0) \simeq 1.1415$, which may be considered as amplifying too much the frequencies around α_0. So we suggest to use, for $n = 1$, $b_1 = -1/4$ which lies between the truncated development case and the least squares one, and gives improved results.

3.2.2. Figures:

Figures 5 to 8 are for quasi-interpolating B-spline approximation, with the least squares criterion, each one for $n = 1, 2, 3, 4$ and 5 (for $n = 1$, $b_1 = -1/4$)

Figure 5: transfer function $H_n(\alpha)$

Figure 6: transfer function $H_n(\alpha)$ (enlarging the scale around $H_n(\alpha)$).

Figures 7 and 8: response to the same data as figures 3 and 4.

When comparing figures 5-8 with figures 1-4, we can see how much closer the least squares B-spline approximation is, for the same n, to the interpolant than the truncated development B-spline approximation is.

3.3 A pretty filter

It is worthwhile to mention the filter defined by its a_j^n coefficients (see section 1) by: $\forall j \in [-n,n]$, $a_j^n = (n + 1 - |j|)/(n + 1)^2$. For lack of space, it is not possible to detail its properties, but we can say that it is a very efficient and simple high-cut filter. Its transfer function is shown at figure 9, for $n = 5, 15, 45$; the curve obtained by filtering some noisy data (shares of some French company) is shown in figure 10.

4. Conclusion

4.1. **"Transfer function" is a most efficient way** to obtain a global view of an approximation process. For quasi-interpolation, it is much better than "an order of convergence h^q" (or "interpolating polynomials of degree at most $q - 1$").

4.2. We think that this algorithm may have many applications:

- in CAGD: thanks to this algorithm, an operator has no need to give points far from the desired curve, which is much more convenient
- for closed curves: the way to extend the points in a periodic way is obvious, and there is of course no linear system to solve.
- for filtering data: most methods need to solve large linear systems; here we get satisfactory results with little computation.
- for d-dimensional surfaces obtained by tensor product: simple

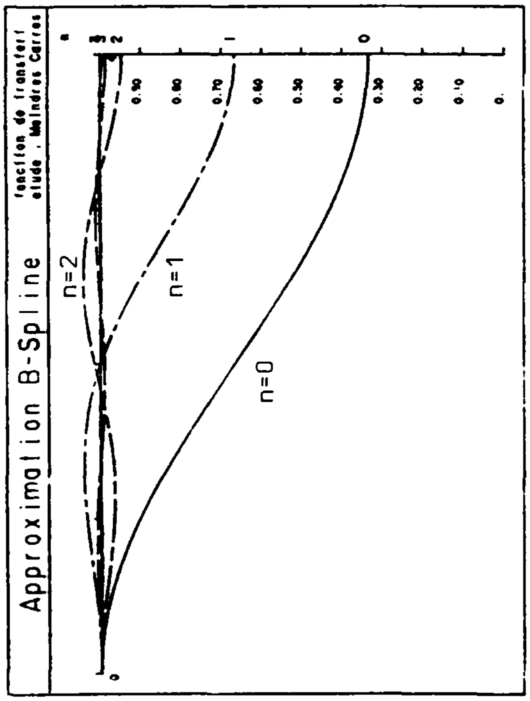


Figure 5

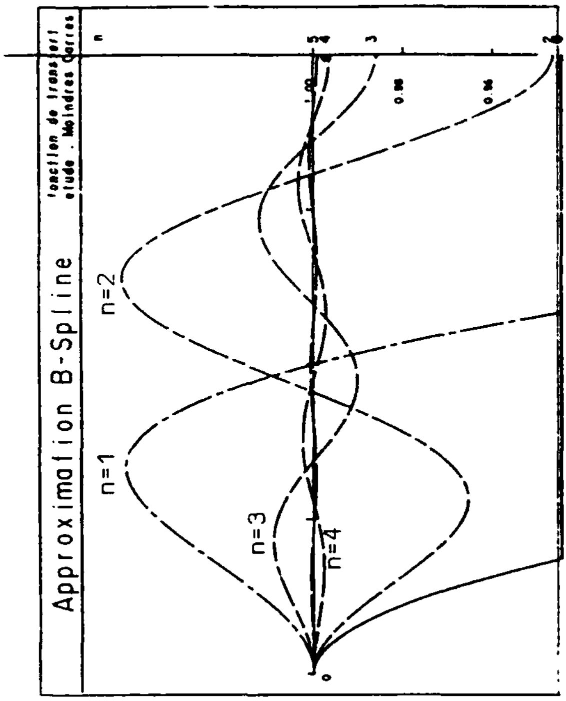


Figure 6

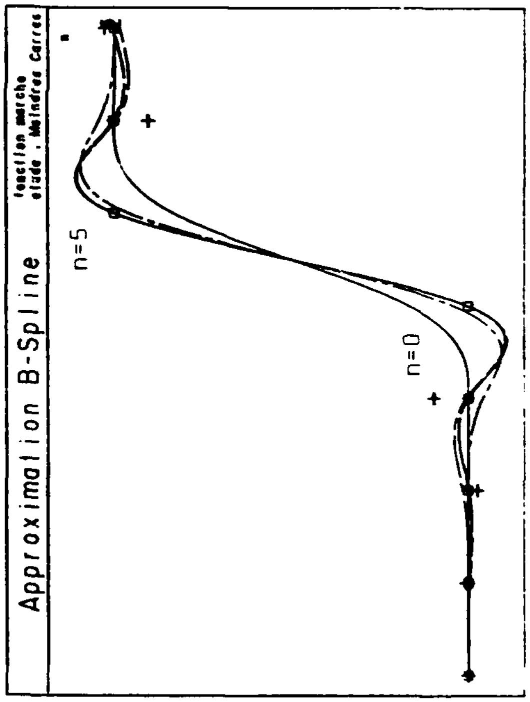


Figure 7

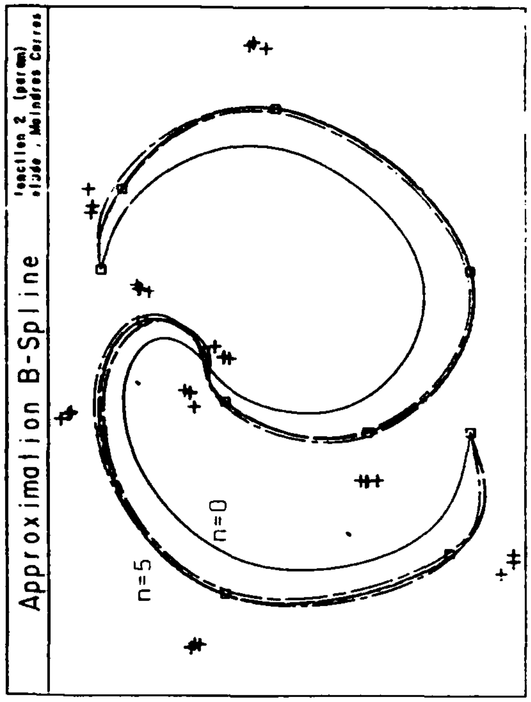


Figure 8

Figures 5 to 8 : Least squares criterion.

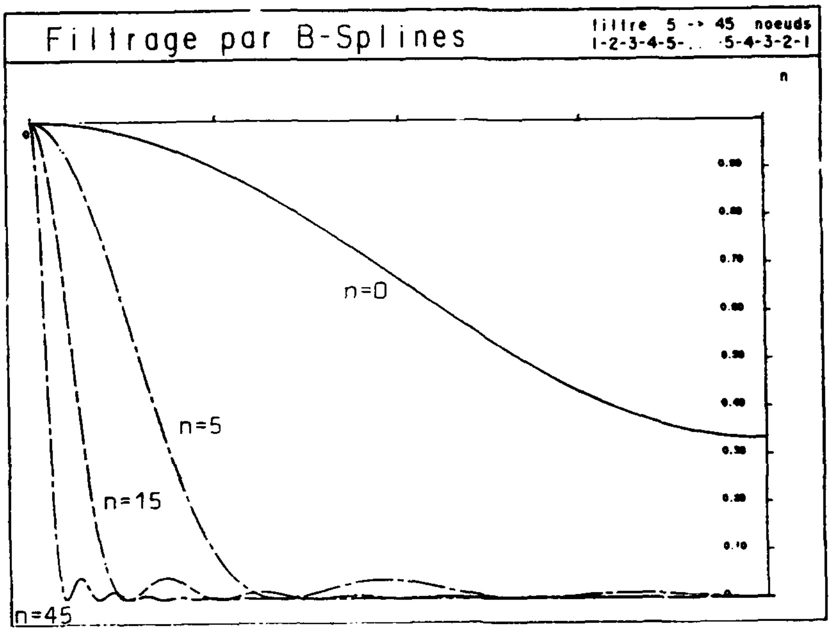


Figure 9

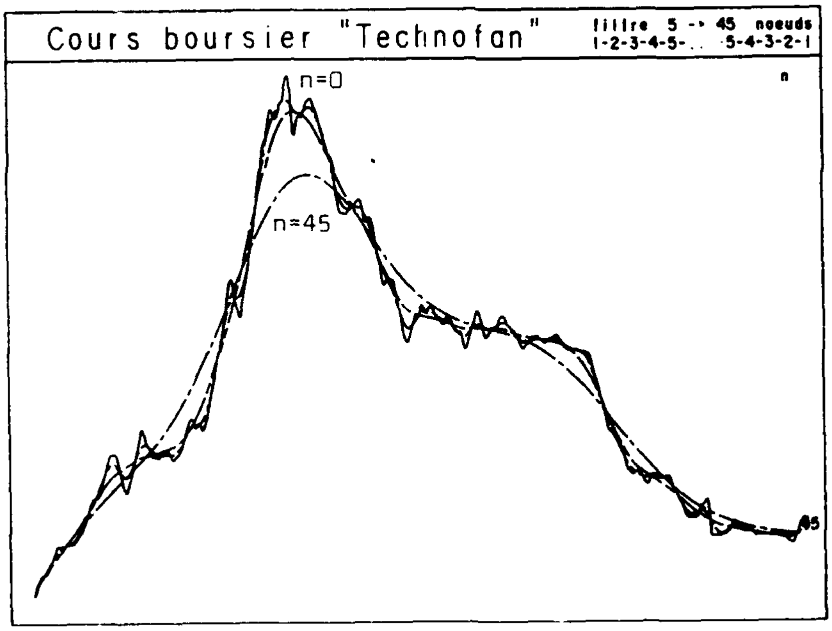


Figure 10

Figures 9 and 10 : A very efficient filter.

quasi-interpolation is very interesting as the usual tensor product B-spline approximation smoothes the data in a quite important way, which of course is not the case of tensor product B-spline quasi-interpolation.

4.3. This method is very easy to introduce in existing algorithms: we only have to add a line to compute the nodes Y_i^n; everything else remains unchanged!

4.4. The "modified B-splines" B^n can certainly be used in place of the usual B-splines in every application (as rational splines). So, we can have the opportunity of one (or even many) degree of freedom.

4.5. At present work is carried out for determining best high-cut filter by cross-validation techniques. Extensions are also being made to thin plate splines.

References

de Boor, C "A practical guide to splines", Springer Verlag 1978

Lyche, T "Local spline approximation methods and osculatory formulae", Approximation Theory, R Schabak and K Scherer (editors), Springer lecture notes in mathematics, 556 (1976), 305-319

Schumaker,L "Spline functions: Basic Theory" 1981, John Wiley and sons

ON KNOTS AND NODES FOR SPLINE INTERPOLATION

PHILIP W. SMITH
IMSL

Abstract This paper explores the relationships among the knots, nodes, and norm of the interpolating spline projection. In particular, we derive lower bounds for the norm of the projection in terms of the knots and nodes. We then turn our attention to the role played by equioscillating splines in deriving the best interpolating projector for a fixed spline subspace.

Keywords: equioscillating spline, B-spline, projection.

1. Introduction

The general problem of choosing knots and/or nodes for spline interpolation is addressed in this paper. Controlling the size of the interpolating projection is the criterion for 'good' knots and/or nodes. We adhere to the following notation throughout this work. The spline subspace S_t^k, $t := t_1 \le \cdots \le t_{n+k}$, is the linear span of the n normalized B-splines $B_{1,k,t}, \dots, B_{n,k,t}$ where

$$B_{i,k,t}(x) := (t_{i+k} - t_i)[t_i, \dots, t_{i+k}]_s (s - x)_+^{k-1}.$$

If no misunderstanding can arise, we denote $B_{i,k,t}$ by B_i. Let $\tau := \tau_1 < \cdots < \tau_n$ be n nodes of interpolation. The operator

$$P : C[t_1, t_{n+k}] \to S_t^k$$

is well defined by

$$(Pf)(\tau_i) = f(\tau_i) \quad i = 1, \dots, n$$

if and only if $B_i(\tau_i) \ne 0$ (i.e. usually $t_i < \tau_i < t_{i+k}$; however, equality can occur if k knots are stacked together). Throughout, we assume that P is well defined.

The central issue is the relationship of the norm of the operator P, $\|P\|$, to the knots and nodes. Instead of working directly with P, we shift our attention to the finite dimensional linear operator $A : R^n \to R^n$ defined by

$$A_{ij} = (A_{n,\tau,t})_{ij} = B_{j,k,t}(\tau_i) \quad 1 \le i,j \le n.$$

The relationship between P and A is

$$Pf = \sum_{j=1}^{n} [A^{-1} f(\tau)]_j B_j,$$

where $f(\tau) := (f(\tau_1), \ldots, f(\tau_n))$. Furthermore, one observes [1] that

$$D_k^{-1} \|A^{-1}\| \le \|P\| \le \|A^{-1}\|. \tag{1}$$

where all norms in this paper are supremum norms on the appropriate space. The important feature of the above inequality is that it is independent of n or t.

Recall that A is totally positive. This means that A^{-1} is checkerboard, that is

$$(-1)^{i+j} A_{ij}^{-1} \ge 0,$$

from which it follows that

$$\|A^{-1}\| = \|A^{-1} e\|$$

where $e_i = (-1)^{i-1}$, for $i = 1, \ldots, n$. From the inequality (1), we see that upper and lower bounds on $\|A^{-1}\|$ in terms of t and τ yield similar bounds on $\|P\|$.

We focus on two problems. The first problem is fundamental but extremely difficult: Given k and τ, choose the 'best' knots for interpolation. We interpret this to mean, find knots t^* which satisfy

$$\|A_{t^*}^{-1}\| \le \|A_t^{-1}\|$$

among all competing knot sequences t. Perhaps it would be even more practical to find 'acceptable' t^{**} which depend on local information and satisfy

$$M_k \|A_{t^{**}}^{-1}\| \le \|A_{t^*}^{-1}\|$$

for some fixed positive constant M_k.

The second problem is, in some sense, the dual problem: Given k and t, choose the best τ for interpolation. We interpret this to mean, find nodes τ^* so that

$$\|A_{\tau^*}^{-1}\| \le \|A_\tau^{-1}\|$$

among all competing nodes τ. S. Demko [3] has shown that there exists τ^{**} so that

$$\|A_{\tau^{**}}^{-1}\| \le D_k.$$

In fact, it has been noted by Morken [4] that Demko's choice of nodes yields the best set of nodes for the given knot sequence.

2. Estimates for $\|A^{-1}\|$

In this section, we elaborate on a technique for obtaining lower estimates for $\|A^{-1}\|$ that first appeared in [1]. Recall [2], that if $s \in S_t^k$ and

$$s = \sum_j b_j B_{j,k}$$

then (ignoring boundary effects)

$$s' = (k-1)\sum_j \frac{b_j - b_{j-1}}{t_{j+k-1} - t_j} B_{j,k-1}.$$

We define the weighted difference operator $\nabla^{(1)}_{t,k}$ by

$$\nabla^{(1)}_{t,k} b_j := (k-1)\frac{b_j - b_{j-1}}{t_{j+k-1} - t_j}$$

which allows us to rewrite s' as

$$s' = \sum_j \nabla^{(1)}_{t,k} b_j B_{j,k-1}.$$

More generally, if we take the rth derivative then

$$s^{(r)} = \sum_j \nabla^{(r)}_{t,k} b_j B_{j,k-r}.$$

where we define $\nabla^{(r)}_{t,k}$ recursively by

$$\nabla^{(r)}_{t,k} = \nabla^{(1)}_{t,k-r+1} \nabla^{(r-1)}_{t,k}.$$

The goal in this section is to estimate the coefficients of an oscillating spline and to relate this information to bounds on $\|A^{-1}\|$. If the spline s satisfies $s(\tau_i) = (-1)^{i-1}$ for $i = 1, \dots, n$, then the ℓ_∞ norm of the coefficients is $\|A^{-1}\|$.

Theorem 1. Let k, n, t, and τ be given. We assume that

$$A = A_{n,t,\tau}$$

is invertible and that $S_t^k \in L_\infty^r[\tau_1, \tau_n]$. Set $e(i) := (-1)^{i-1}$ for $i = 1, \dots, n$. Then

$$\|A^{-1}\| \geq \frac{|[\tau_i, \dots, \tau_{i+r}]e|}{\Delta_{i,r}} r!$$

where $i = 1, \dots, n-r$ and

$$\Delta_{i,r} := \max\{|\nabla^{(r)}_{t,k} e_j| : (t_j, t_{j+k-r}) \cap (\tau_i, \tau_{i+r}) \neq \emptyset\}.$$

Proof: Let $s \in S_t^k$ satisfy

$$s(\tau_i) = e_i = (-1)^{i-1} \quad i = 1, \dots, n.$$

Then $s = \sum_j a_j B_{j,k}$ and $Aa = e$. It follows that $\|A^{-1}\| = \|a\|$. From the Peano kernel theorem we have

$$[\tau_i, \dots, \tau_{i+r}]s = [\tau_i, \dots, \tau_{i+r}]e \leq \frac{\|s^{(r)}\|_{[\tau_i,\tau_{i+r}]}}{r!}.$$

Now

$$s^{(r)} = \sum_j \nabla_{t,k}^{(r)} a_j B_{j,k-r}$$

and hence

$$\|s^{(r)}\|_{[\tau_i,\tau_{i+r}]} \leq \max\{|\nabla_{t,k}^{(r)} a_j| : \text{supp } B_{j,k-r} \cap (\tau_i, \tau_{i+r}) \neq \emptyset\}.$$

Since the matrix A is variation diminishing, we know that $a_j(-1)^{j-1} \geq 0$ for all j. It follows that there is no cancellation in the differences $\nabla_{t,k}^{(r)} a_j$, and hence

$$|\nabla_{t,k}^{(r)} a_j| \leq |\nabla_{t,k}^{(r)} e_j| \|a\|.$$

Assembling the pieces yields

$$|[\tau_i, \dots, \tau_{i+r}]e| \leq \|a\| \frac{\Delta_{i,r}}{r!},$$

completing the proof.

We mention two corollaries of this result.

Corollary 2. (de Boor [1]) Let $r = 1$, then

$$\|A^{-1}\| \geq \frac{d_i}{(\tau_{i+1} - \tau_i)(k-1)}$$

and

$$\|P\| \geq \frac{D_k d_i}{(\tau_{i+1} - \tau_i)(k-1)}.$$

where

$$d_i := \min\{t_{j+k-1} - t_j : (t_j, t_{j+k-1}) \cap (\tau_i, \tau_{i+1}) \neq \emptyset\}$$

The second inequality was derived by Ching-Ching Rojas, a student of ours.

Corollary 3. Let $r = 2$ and $k = 3$ with $S_t^k \subset L_\infty^2$. Then

$$\|A^{-1}\| \geq \frac{\delta_i}{(\tau_{i+2} - \tau_{i+1})(\tau_{i+1} - \tau_i)}$$

where

$$\delta_i = \max\{\frac{(t_{j+1} - t_{j-1})(t_{j+2} - t_j)(t_{j+1} - t_j)}{(t_{j+1} - t_{j-1}) + (t_{j+2} - t_j)} : (t_j, t_{j+1}) \cap (\tau_i - \tau_{i+2})\}.$$

3. Equioscillating Splines

In [3] it was shown that there exists a spline $T = \sum a_j^* B_{j,k} \in S_t^k$ which equioscillates maximally. That is, on $[t_1, t_{n+k}]$ there exist n points $\tau_1 < \tau_2 < \cdots < \tau_n$ so that

$$T(\tau_i) = (-1)^{i-1}$$
$$\|T\| = 1.$$

Combining this observation with the inequality

$$D_k^{-1}\|a^*\| \leq \|T\| = 1$$

yields the following result

Theorem 4. (Demko [3]) Given n, t, and k, there exist nodes τ so that the matrix $A := A_{n,t,\tau}$ satisfies

$$\|A^{-1}\| \leq D_k.$$

It is important to note that this result yields a universal upper bound (depending only on k) for these particular nodes. Indeed, this result illustrates a dichotomy between the dual problems of obtaining

$$\text{(A)} \quad \sup_t \inf_\tau \|A_{t,\tau}^{-1}\|$$
$$\text{(B)} \quad \sup_\tau \inf_t \|A_{t,\tau}^{-1}\|$$

Note that by Theorem 3.1, (A) is bounded above by D_k, while (B) is always infinite by Theorem 2.1 or Corollary 2.2, if $k > 2$ and n is sufficiently large.

As Morken [4] pointed out, the abscissae of the extreme points of the equioscillating spline yield the best points for interpolation. This can be seen by setting T to be the equioscillating spline with equioscillation points τ, as above, and letting $\zeta_1 < \cdots < \zeta_n$ be another set of nodes. Let $s \in S_t^k$ be defined by $s(\zeta_i) = (-1)^{i-1}$, then

$$T = \sum a_j^* B_{j,k}$$
$$s = \sum b_j B_{j,k}$$
$$(s - T)(\zeta_i)(-1)^{i-1} \geq 0.$$

This means that the coefficients of $s - T$, which are $b_j - a_j^*$, weakly oscillate (i.e. $(b_j - a_j^*)(-1)^{j-1} \geq 0$). But since b_j and a_j^* oscillate in the same orientation, we conclude that

$$|a_j^*| \leq |b_j| \quad j = 1, \ldots, n$$
$$\text{hence } \|a^*\| \leq \|b\|$$
$$\text{or } \|A_\tau^{-1}\| \leq \|A_\zeta^{-1}\|$$

We now discuss the computing of the nodes τ for the equioscillating spline. First, observe that T (and hence τ) is unique, since T can be viewed as an error in the best approximation of $a_1^* B_{1,k}$ from the span of $\{B_{2,k}, \dots, B_{n,k}\}$. Furthermore, if we set $t_1 = \cdots = t_k$ and $t_{n+1} = \cdots = t_{n+k}$, then $\tau_1 = t_k$ and $\tau_n = t_{n+1}$. This follows by noting that all B-splines vanish at t_k except for $B_{1,k}$, and this spline takes on its maximum at t_k. Similarly, all B-splines vanish at t_{n+1} except $B_{n,k}$, which takes on its maximum at t_{n+1}.

The computational algorithm is a Remez exchange algorithm, where τ^1 is chosen so that A_{t,τ^1} is invertible and $\tau_1^1 = t_k$, $\tau_n^1 = t_{n+1}$. Currently, we choose τ to be the interior knot averages for the B-splines, that is

$$\tau_i := \frac{\sum_{j=i+1}^{i+k-1} t_j}{k-1}.$$

We then compute s_1 which satisfies

$$s_1(\tau_i^1) = e_i = (-1)^{i-1}.$$

We then choose τ^2 to be the unique strictly increasing vector satisfying

$$\begin{aligned} \tau_1^2 &= t_k \\ \tau_n^2 &= t_{n+1} \\ s_1'(\tau_j^2) &= 0 \quad j = 2, \dots, n-1. \end{aligned}$$

Then τ^2 is substituted for τ^1, and the process is repeated until convergence.

References

1. Carl de Boor. On bounding spline interpolation. *J. of Approximation Theory*, 14:191–203, 1975.
2. Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.
3. S. Demko. On the existence of interpolating projections onto spline spaces. *J. of Approximation Theory*, 43:151–156, 1985.
4. Knute Morken. *On Two Topics in Spline Theory: Discrete Splines and the Equioscillating Spline*. Master's thesis, Institute for Informatics, University of Oslo, 1984.

2. Polynomial & Piecewise Polynomial Approximation

A BASIS FOR CERTAIN SPACES OF MULTIVARIATE POLYNOMIALS AND EXPONENTIALS

W. DAHMEN
Fachbereich Mathematik
Freie Universität Berlin

Abstract A basis for certain spaces of multivariate polynomials and exponentials is constructed from the polynomial pieces of so-called multivariate truncated powers introduced by Dahmen (1980). The need for such a basis arises in connection with certain Hermite interpolation problems which were recently investigated by Dyn and Ron (1988a, b). It is indicated how to compute the elements of the basis explicitly by means of various representations of truncated powers.

Key words: Truncated powers, Exponential splines, Interpolation, Construction of bases, Common null space of families of differential operators.

1. Introduction

The literature of the past few years reflects a rapidly growing interest in multivariate interpolation problems. The numerous methods which have been developed so far are based on a diversity of function systems where an important role is played, of course, by spaces of splines and polynomials.

This paper is concerned with certain spaces of multivariate polynomials and exponentials which are obtained as common null spaces of certain differential operators (see e.g. Dahmen and Micchelli (1983a), (1985), (1987)). These spaces play a central role in the theory of multivariate splines on regular grids as well as in various related combinatorial and algebraic aspects (cf. Dahmen and Micchelli (1988)). Moreover, they arise in connection with certain Hermite interpolation problems that were recently investigated by Dyn and Ron (1988a, 1988b).

However, the practical solution of such interpolation problems would require the explicit knowledge of bases for these spaces which it seems have not previously been available.

The objective of this paper is to construct such bases by making use of the intimate relationship to the corresponding multivariate splines. More precisely, it will be shown how to obtain a basis from the pieces of certain splines called multivariate truncated powers introduced by Dahmen (1980). The computation of these pieces is facilitated by various known representations for truncated powers.

The paper is organized as follows. In Section 2 we state the relevant definitions and collect some background material. Section 3 is devoted to the general construction of bases for the above mentioned spaces. Finally, an example is discussed in Section 4.

2. Some definitions and background material

Let us start by recalling the definitions and some properties of certain multivariate (polynomial and exponential) splines which will play a crucial role in subsequent discussions. For more technical details and proofs the reader is referred to Dahmen (1980), Dahmen and Micchelli (1983b), (1987). To this end, let $X = \{x^1,\dots,x^n\}$ be a set of (not necessarily distinct) vectors in $\mathbb{R}^s\setminus\{0\}$ and let $\mu \in \mathbb{C}^n$ be fixed. For convenience the $(s \times n)$-matrix whose columns are the elements of X is also denoted by X. Under suitable assumptions on μ and X one can then define a function $T_\mu(\cdot|X)$ by requiring that

$$\int_{\mathbb{R}^s} f(x)T_\mu(x|X)dx = \int_{\mathbb{R}^n_+} e^{-\mu\cdot v} f(Xv)dv \tag{1}$$

holds for all $f \in C_0(\mathbb{R}^s)$ (where $v \cdot z = v_1\bar{z}_1 + \ldots + v_n\bar{z}_n$ denotes the standard inner product on $\mathbb{C}^n$). A case of particular interest is $\mu = 0$. For $T(\cdot|X) := T_0(\cdot|X)$ to be well defined one has to assume that

$$0 \notin [X] , \tag{2}$$

where $[X]$ denotes the convex hull of the set X. If $\langle X\rangle$, the span of X, is all of $\mathbb{R}^s$, $T(\cdot|X)$ is easily seen to be indeed a function with support

$$\operatorname{supp} T(\cdot|X) = \langle X\rangle_+ , \tag{3}$$

where $\langle X\rangle_+ = \{Xv : v \in \mathbb{R}^n_+\}$ is the cone spanned by X. $T(\cdot|X)$ can be shown to be a piecewise polynomial of degree at most $n - s$, where the polynomial pieces are separated by the hyperplanes in

$$C(X) = \{\langle V\rangle : V \in B_{s-1}(X)\} . \tag{4}$$

Here

$$B_\ell(X) = \{V \subset X : \dim\langle V\rangle = |V| = \ell\}$$

and $|V|$ denotes the cardinality of V. In general, when μ is some complex vector one has to assume

$$\operatorname{Re} \mu_j > 0 \tag{5}$$

whenever x^j belongs to X', the largest subset of X such that there exists $\beta \in \mathbb{Z}_+^{|X'|}$, $\beta_i > 0$, $i = 1,\ldots,|X'|$ with $X'\beta = 0$. Again $T_\mu(\cdot|X)$ turns out to be piecewise analytic with cut regions given by (4) and support $\langle X\rangle_+$. In general its pieces are composed of polynomials and exponentials to be explained in more detail below. Specifically, one readily concludes from (1) that for $n = |X| = s$,

$<X> = \mathbb{R}^s$,

$$T_\mu(x|X) = \chi_{<X>_+}(x)\,|\det X|^{-1}e^{-\mu\cdot(X^{-1}x)}. \qquad (6)$$

In general $T_\mu(\cdot|X)$ can be evaluated recursively. In addition, explicit analytic expressions are available for special choices of X. The following case will be needed in the next section. To this end, let

$$X = \{y^1,\ldots,y^1,\ldots,y^s,\ldots,y^s\},$$

where each y^i occurs m_i times and $Y = \{y^1,\ldots,y^s\}$ spans $\mathbb{R}^s$. Denoting by $u = Y^{-1}x$ the coordinates of x with respect to the basis Y one has (cf. Dahmen and Micchelli (1983b))

$$T(x|X) = \chi_{<Y>_+}(x)u_1^{m_1-1}\ldots u_s^{m_s-1}. \qquad (7)$$

When $\mu \neq 0$ note first that for any nonsingular $(s \times s)$-matrix A:

$$|\det A|T_\mu(Ax|AX) = T_\mu(x|X).$$

Choosing $A = Y^{-1}$ one then easily verifies that

$$T_\mu(x|X) = |\det Y|T_\mu(u|Y^{-1}X) = \prod_{j=1}^{s} T_{\mu^j}(u_j|1(j)), \qquad (8)$$

where $\mu^j = (\mu_{m_1+\ldots+m_{j-1}+1},\ldots,\mu_{m_1+\ldots+m_j})$ and $1(j)$ denotes the $(1 \times m_j)$-matrix with all entries equal to 1. Moreover, noting that the Laplace transform of each factor $T_{\mu^j}(\cdot|1(j))$ is given by $\Pi_{i=1}^{m_j}(\mu_i^j + \lambda)^{-1}$, $\lambda > -\mu_i^j$, $i = 1,\ldots,m_j$, each univariate factor in (8) is a convolution of the functions $t_+^0\, e^{-\mu_i^j t}$. Defining $D_y f = \Sigma_{j=1}^s y_j \frac{\partial f}{\partial x_j}$ and denoting by $\mu\setminus j$ the element in $\mathbb{C}^{n-1}$ obtained from μ by discarding μ_j it is also not hard to show that

$$(\mu_j + D_{x^j})T_\mu(\cdot|X) = T_{\mu\setminus j}(\cdot|X\setminus\{x^j\}). \qquad (9)$$

Many of the properties stated above indeed follow from (9). In particular, setting

$$\mathcal{Y}(X) = \{V \subset X : \langle X \setminus V \rangle \neq \mathbb{R}^s\},$$

a repeated application of (9) yields, in view of (6), that

$$(\Pi_{v \in V}(\mu_v + D_v))T_\mu(x|X) = 0 \tag{10}$$

for all $V \in \mathcal{Y}(X)$, provided that x does not belong to any of the cut regions in (4). Hence $T_\mu(\cdot|X)$ belongs locally to the space

$$D_\mu(X) = \{f \in \mathcal{D}'(\mathbb{R}^s) : (\Pi_{v \in V}(\mu_v + D_v))f = 0, \forall V \in \mathcal{Y}(X)\}.$$

where $\mathcal{D}'(\mathbb{R}^s)$ denotes the space of Schwartz distributions on $\mathbb{R}^s$. The fact that $D_\mu(X)$ is finite dimensional is a consequence of the following more general result.

Theorem 1 Suppose that for X as above $\{L_v\}_{v \in X}$ is a family of commuting endomorphisms on some linear space S . Setting

$$K(X) = \{f \in S : (\Pi_{v \in V} L_v)f = 0 \ , \ V \in \mathcal{Y}(X)\},$$

one has

$$\dim K(X) \leq \Sigma_{Y \in \mathcal{B}_s(X)} \dim K(Y) . \tag{11}$$

Theorem 1 is a special case of Theorem 3.1 in Dahmen and Micchelli (1987). Moreover, conditions are given there which ensure that equality holds in (11). Specializing these facts to the case at hand gives

$$\dim D_\mu(X) = |\mathcal{B}_s(X)| \tag{12}$$

independent of $\mu \in \mathbb{C}^n$.

Defining for $Y = \{x^{i_1},\ldots,x^{i_s}\} \in \mathcal{B}_s(X)$,

$$u_Y = Y^{-T}\mu_Y , \tag{13}$$

where $\mu_Y = (\mu_{i_1},\ldots,\mu_{i_s})$, the common zeros of the polynomials $\Pi_{v \in V}(x \cdot v + \mu_v)$, $V \in \mathcal{V}(X)$, are the points $- u_Y$, $Y \in \mathcal{B}_s(X)$. Hence one expects that $D_\mu(X)$ is spanned by functions of the form $\exp\{- u_Y \cdot x\}p(x)$, where $p(x)$ is some polynomial. More precisely, let us call any two elements Y , $Y' \in \mathcal{B}_s(X)$ equivalent if and only if $u_Y = u_{Y'}$ and let $\mathcal{E}_j$, $j = 1,\ldots,m$, denote the corresponding equivalence classes with representers $Y_j \in \mathcal{E}_j$, $j = 1,\ldots,m$. Moreover, define for any $Y \in \mathcal{B}_s(X)$

$$X_Y = \{z \in X : z \cdot u_Y = \mu_z\} ,$$

i.e. $Y \subseteq X_Y$ so that $\langle X_Y \rangle = \mathbb{R}^s$. It is pointed out in Dahmen and Micchelli (1987) and Ben-Artzi and Ron (1987) that when $p_{j,i}$, $i = 1,\ldots,\ell_j$, is a basis of $D_0(X_{Y_j})$ then the collection of functions

$$\{\exp(- u_{Y_j} \cdot x)p_{i,j}(x) : i = 1,\ldots,\ell_j , j = 1,\ldots,m\} \tag{14}$$

forms a basis for $D_\mu(X)$.

The spaces $D_0(X)$, $D_\mu(X)$ play a fundamental role for the theory of box or cube splines and exponential cube splines (cf. Dahmen and Micchelli (1987), de Boor and Höllig (1982/83)) leading among other things to interesting implications concerning related combinatorial and algebraic problems (see Dahmen and Micchelli (1988), (1987)). Moreover, certain interpolation problems were recently shown by Dvn and Ron (1988a), (1988b) to have unique solutions in these **spaces.**

To make any practical use of these results requires the determination of appropriate bases for these spaces. In view of (14) this reduces to constructing for any X a basis for $D_0(X)$. Since we will be mainly concerned with this space we will henceforth drop the subscript 0. It was shown by Dahmen and Micchelli (1983a) that D(X) contains only polynomials. In fact, one has the inclusion

$$\Pi_{d(X)}(\mathbb{R}^s) \subseteq D(X) \subseteq \Pi_{n-s}(\mathbb{R}^s), \tag{15}$$

where $d(X) + 1 = \min\{|V| : V \in \mathcal{Y}(X)\}$ and $\Pi_k(\mathbb{R}^s)$ denotes the space of all real polynomials of total degree at most k on $\mathbb{R}^s$. Although the dimension of D(X) is known precisely, its structure is generally rather complicated. For instance, when $s = 2$, $X = \{(1,0),(1,0),(0,1),(1,1)\}$, one easily verifies that $\{1,x,y,x^2,y^2 - 2xy\}$ is a basis for D(X) in this case. However, in general, when more directions are involved or when dealing with more than two variables no general recipe for constructing a basis seems to be known. Exploiting the intimate connection between the space D(X) and certain spline spaces a general strategy for constructing a basis for D(X) will be derived in the following section.

3. A basis for D(X)

It was pointed out by Dahmen and Micchelli (1983a) and deBoor and Höllig (1982/83) that when $X \subseteq \mathbb{Z}^s\setminus\{0\}$ the space D(X) is spanned by the translates of the cube spline $C(\cdot|X)$. If, in addition, X satisfies (2) then $C(\cdot|X)$ is related to the truncated power by

$$C(\cdot|X) = \nabla_X T(\cdot|X),$$

where $\nabla_y f(\cdot) = f(\cdot) - f(\cdot - y)$ and for $v \in V$, $\nabla_V f = \nabla_v(\nabla_{V\setminus\{v\}} f)$. Hence appropriate pieces of multivariate truncated powers should be expected to form a basis of D(X). It will be shown next that this is indeed the case even for arbitrary $X \subset \mathbb{R}^s\setminus\{0\}$.

Note first that replacing x^j in X by cx^j for any $c \in \mathbb{R}$ will leave D(X) unchanged. Thus, assigning appropriate signs to the elements of X , we may assume without loss of generality that X satisfies (2) so that $T(\cdot|X)$ is well defined. Moreover, upon rescaling the x^i if necessary we may assume throughout the following that

$$\langle x^i\rangle = \langle x^j\rangle \text{ implies } x^i = x^j . \qquad (16)$$

When $|X| = n = s$ there is nothing to prove since $D(X) = \Pi_0(\mathbb{R}^s)$ This suggests proceeding by induction on $|X|$. Suppose that for some X, $|X| = n \geq s$ one has already determined a basis for $D(X)$. Furthermore, suppose that

$$X' = X \cup \{y\}$$

also satisfies (2). Then, defining

$$A(X|y) = \{Y \in \mathcal{B}_{s-1}(X) : \{y\} \cup Y \in \mathcal{B}_s(X')\} ,$$

one has

$$|\mathcal{B}_s(X')| = |A(X|y)| + |\mathcal{B}_s(X)| . \qquad (17)$$

Thus it remains to construct $N:=|A(X|y)|$ additional linearly independent elements in the quotient space $D(X')/D(X)$.

The construction of these additional polynomials will be based on an appropriate ordering for $A(X_0|y)$, where $X_0 \subseteq X$ is a maximal subset of pairwise distinct elements in X.

To this end, $S = \{S_j : j = 1,\ldots,L\}$ will denote a fixed maximal sequence in $A(X_0|y)$ such that the cones $C_j = \langle S_j \cup \{y\}\rangle_+$ satisfy

$$C_j \setminus \cup_{i<j} C_i \neq \emptyset . \qquad (18)$$

Without loss of generality one can assume that no element of S can be written as the union of essentially disjoint cones of the form $\langle Y \cup \{y\}\rangle_+$, $Y \in A(X_0|y)$. In fact, if such a cone C belongs to S at least one of the cones C' contained in C cannot belong to S. Swapping the corresponding elements Y, Y' of $A(X_0|y)$ would preserve (18) for a sequence of at least the same length L.

The following information about the elements of $\mathcal{A}(X_0|y)$ which are not contained in $\mathcal{S}$ will be useful.

Lemma 1 For every $Y \in \mathcal{A}(X_0|y) \setminus \mathcal{S}$ there exists a subset $\mathcal{H}$ of $\mathcal{S}$ such that

$$\langle Y\rangle_+ \subseteq \cup\{\langle S\rangle_+ : S \in \mathcal{H}\} ,$$

i.e. $\langle Y\rangle = \langle S\rangle$, $S \in \mathcal{H}$.

Proof Let $j \leq L$ be the smallest index such that

$$C = \langle\{y\} \cup Y\rangle_+ \subseteq \cup_{i \leq j} C_i , \tag{19}$$

i.e.

$$C \setminus \cup_{i<j} C_i \neq \emptyset . \tag{20}$$

In particular, this means that

$$K = \langle Y\rangle_+ \setminus \cup_{i<j} C_i \neq \emptyset$$

and $K \subset C_j$. If K intersects the interior of C_j one easily concludes that

$$C_j \setminus ((\cup_{i<j} C_i) \cup C) \neq \emptyset .$$

In view of (20) this implies that inserting Y between S_{j-1} and S_j still provides a sequence satisfying (18) and thus contradicting the maximality of $\mathcal{S}$. Hence $K \subseteq \langle s_j\rangle_+$ which means $\langle s_j\rangle = \langle Y\rangle$. Let $v \in S_j \setminus Y$. Clearly $\langle Y\rangle_+ \subseteq \cup\{\langle\{v\} \cup (Y \setminus \{z\})\rangle_+ : z \in Y\}$. Since every $\langle V\rangle_+$, $V \in \mathcal{A}(X_0|y)$, is covered by (s - 1)-cones $\langle S\rangle_+$, $S \in \mathcal{S}$, the assertion follows.

An immediate consequence of Lemma 1 is that $\mathcal{S} = \mathcal{A}(X_0|y)$ if the vectors in X_0 are in general position. In particular, this is the case for any $X \subset \mathbb{R}^2$ satisfying (2).

Now let $\mathcal{G}_j$ denote the (possibly empty) set of all $Y \in \mathcal{A}(X_0|y) \setminus \mathcal{S}$ such that the highest index j of a covering $\mathcal{H}$ for $<Y>_+$ given by Lemma 1 is as small as possible. Fixing any ordering for $\mathcal{G}_j$ and inserting $\mathcal{G}_j$ between S_j and S_{j+1} establishes an ordering for all of $A(X_0|y)$ which will be denoted throughout the sequel by $A(X_0|y) = \{Y_j : j = 1,\dots,M\}$. The corresponding cones $<Y_j \cup \{y\}>_+$ will again be denoted by C_j.

The role of $A(X_0|y)$ with respect to the full set $A(X|y)$ becomes clear when considering the equivalence relation

$$Y \sim V \quad \text{iff} \quad <Y \cup \{y\}>_+ = <V \cup \{y\}>_+ \;,\; Y, V \in A(X|y).$$

In fact, recalling (16), one easily verifies that $A(X_0|y)$ is a set of representers for the corresponding equivalence classes $\mathcal{E}_j$ of $Y_j \in A(X_0|y)$. Note that for $Y_i = \{x^{i,1},\dots,x^{i,s-1}\}$ the cardinalities $\ell_i = |\mathcal{E}_i|$ are given by

$$\ell_i = m_{i,1} \cdots m_{i,s-1} , \tag{21}$$

where $x^{i,k}$ occurs $m_{i,k}$ times in X . For any $I = (i_1,\dots,i_{s-1}) \in \mathbb{Z}_+^{s-1}$ let

$$Y_i(I) = \{x^{i,1},\dots,x^{i,1},\dots,x^{i,s-1},\dots,x^{i,s-1}\} ,$$

where $x^{i,k}$ occurs exactly i_k times in $Y_i(I)$. Specifically, $Y_i((1,\dots,1)) = Y_i$, while $\hat{Y}_i$ will be used as a shorthand notation for $Y_i((m_{i,1},\dots,m_{i,s-1}))$. Let $\ell(i) \leq i$ be the largest integer such that $Y_{\ell(i)} \in \mathcal{S}$. Let $\Delta_i = <Y_{\ell(i)}>_+ \setminus \cup_{j<\ell(i)} <Y_j>_+$ and define for $Y_i \in \mathcal{S}$

$$H_i = (\cup\{\hat{Y}_j : \ell(i) \leq j < i \,,\, <Y_j> = <Y_i>, <Y_j>_+ \cap \Delta_i \neq \emptyset\} \setminus \hat{Y}_i , \tag{22}$$

while $H_i = \emptyset$ when $Y_i \in \mathcal{S}$. Furthermore, let

$$\mathcal{J}_i = \{I = (i_1,\dots,i_{s-1}) : i_k \in \{0,\dots,m_{i,k}-1\} \,,\, k = 1,\dots,s-1\} , \tag{23}$$

so that $|\mathcal{J}_i| = \ell_i = |\mathcal{E}_i|$. Finally, defining, for $I \in \mathcal{J}_i$,

$$V(I) = Y_i(I) \cup H_i , \tag{24}$$

one is ready to construct additional linearly independent polynomials in $D(X')\setminus D(X)$. Writing briefly $D_V f = (\Pi_{v\in V} D_v) f$ the polynomials $P_{i,I}(x)$, $I \in \mathcal{J}_i$, $i = 1,\dots,M$, are defined to be extensions of certain polynomial pieces of truncated powers, namely

$$P_{i,I}(x) = D_{V(I)} T(x.X') \text{ , } x \in (C_{\ell(i)} \setminus U_{j<\ell(i)} C_j) \cap C_i . \tag{25}$$

The main result of the paper may be stated as follows.

Theorem 2 Let X , $X' = X \cup \{y\}$ satisfy (2) and let $P_{i,I}$ be defined by (25). Then for any basis $\mathcal{B}$ of $D(X)$ the set

$$\mathcal{B} \cup \{P_{i,I} : I \in \mathcal{J}_i \text{ , } i = 1,\dots,M\}$$

is a basis for $D(X')$.

Proof Note that in view of (21), (23) there are exactly

$$\sum_{i=1}^{M} |\mathcal{J}_i| = \sum_{i=1}^{M} |\mathcal{E}_i| = |A(X|y)| = N$$

functions $P_{i,I}$. In view of (12) and (17) it remains to show that the $P_{i,I}$ represent linearly independent elements in $D(X')/D(X)$.

To this end, recall from (9) that for any $V \subset X'$,

$$D_V T(\circ|X') = T(\cdot|X'\setminus V) . \tag{26}$$

Since $D(W) \subset D(X)$ for $W \subset X$, (10) readily confirms that

$$P_{i,I} \in D(X') \text{ , } I \in \mathcal{J}_i \text{ , } i = 1,\dots,M . \tag{27}$$

Next observe that for $x \in (C_{\ell(i)} \setminus U_{j<\ell(i)} C_j) \cap C_i$,

$$D_{X\setminus\langle Y_i\rangle} P_{i,I}(x) = D_{X\setminus\langle Y_i\rangle} D_{V(I)} T(x|X')$$

$$= T(x|\{y\} \cup (X \cap \langle Y_i\rangle)\setminus V(I)) .$$

By definition (24) of $V(I)$, the set $(X \cap \langle Y_i\rangle)\setminus V(I)$ still contains some element of $\mathcal{A}(X|y)$ while also (cf. (3))

$$C_i \cap (C_{\ell(i)}\setminus \cup_{j<\ell(i)} C_j) \subseteq \langle\{y\} \cup ((X\cap\langle Y_i\rangle)\setminus V(I))\rangle_+$$

$$= \operatorname{supp} T(\cdot|\{y\} \cup ((X\cap\langle Y_i\rangle)\setminus V(I))) .$$

Hence $D_{X\setminus\langle Y_i\rangle}P_{i,I}$ does not vanish identically. But since $X\setminus\langle Y_i\rangle \in \mathcal{Y}(X)$, this means that

$$P_{i,I} \not\in D(X) \;,\; I \in \mathcal{J}_i \;,\; i = 1,\ldots,M . \qquad (28)$$

So, in view of (27), (28), it remains to confirm the linear independence of the $P_{i,I}$ in $D(X')/D(X)$. To this end, the following observation is useful.

<u>Lemma 2</u> For $i = 1,\ldots,M$, $M = |\mathcal{A}(X_0|y)|$, let

$$W_i = \{y\} \cup V(m^i) ,$$

where $m^i = (m_{i,1},\ldots,m_{i,s-1})$. Then the functions $D_{X'\setminus W_i}P_{i,I}$, $I \in \mathcal{J}_i$, are linearly independent.

<u>Proof</u> Suppose

$$\Sigma_{I\in\mathcal{J}_i} c_I D_{X'\setminus W_i} P_{i,I}(x) = 0 \;,\; x \in \mathbb{R}^s .$$

By definition (25), (26) and (9) this implies

$$\Sigma_{I\in\mathcal{J}_i} c_I T(x|W_i\setminus V(I)) = 0 \;,\; x \in (C_{\ell(i)}\setminus\cup_{j<\ell(i)}C_j) \cap C_i .$$

By (24) one has

$$\{W_i \setminus V(I) : I \in J_i\} = \{\{y\} \cup Y_i \cup Y_i(I') : I' \in J_i\}.$$

Hence the summands $T(\cdot|W_i \setminus V(I))$, $I \in J_i$, range over all different products of the form (7). Since these functions are obviously linearly independent one concludes $c_I = 0$, $I \in J_i$. This completes the proof of Lemma 2.

Next suppose that for a given basis $\{Q_j : j = 1,\dots,|\mathcal{B}_s(X)|\}$ of $D(X)$

$$\Sigma_{i=1}^{M} \Sigma_{I\in J_i} c_{i,I} P_{i,I}(x) + \Sigma_{j=1}^{|\mathcal{B}_s(X)|} a_j Q_j(x) = 0 , x \in \mathbb{R}^s .$$

Since $X' \setminus W_k \in \mathcal{Y}(X)$ one also has for any $k \le M$,

$$\Sigma_{i=1}^{M} \Sigma_{I\in J_i} c_{i,I} D_{X'\setminus W_k} P_{i,I}(x) = 0 , x \in \mathbb{R}^s . \qquad (29)$$

<u>Lemma 3</u> For any $I \in J_i$, $i > k$, one has

$$D_{X'\setminus W_k} P_{i,I}(x) = 0 , x \in \mathbb{R}^s . \qquad (30)$$

<u>Proof</u> Note that for fixed i and $x \in (C_{\ell(i)} \setminus \cup_{j<\ell(i)} C_j) \cap C_i$,

$$D_{X'\setminus W_k} P_{i,I}(x) = D_{V(I)} T(x|X' \setminus (X' \setminus W_k)) \qquad (31)$$

$$= D_{V(I)} T(x|W_k) .$$

Let $k < i$ and suppose first $Y_i \in \mathcal{S}$, i.e. $\ell(i) = i$. Note that in this case

$$C_i \setminus \langle W_k \rangle_+ \neq \emptyset . \qquad (32)$$

In fact, (32) readily follows from (18) and (22) when $\langle Y_i \rangle \neq \langle Y_k \rangle$ or when $Y_k \in \mathcal{S}$. So suppose $\langle Y_i \rangle = \langle Y_k \rangle$ and $Y_k \notin \mathcal{S}$. Let R denote an (s-2)-dimensional supporting hyperplane of Δ_i that separates Δ_i and Δ_k. By

(22) those Y_m that contribute to H_k must be on the same side of R as Δ_k, again confirming (32). Thus, when $Y_i \in \mathcal{S}$ (30) follows from (3), (31) and (32). The same reasoning applies if $Y_i \notin \mathcal{S}$ but $k < \ell(i)$. So assume next that $\ell(i) \leq k < i$. Hence $<Y_i> = <Y_k>$. From (22), (24) one concludes that $\dim<W_k \setminus V(I)> < s$ so that (30) follows again from (3), (31) and (26). This completes the proof of Lemma 3.

Choosing now successively $k = 1,2,\ldots,M$ in (29) and invoking Lemma 2 at each step shows that $c_{i,I} = 0$, $I \in \mathcal{J}_i$, $i = 1,\ldots,M$. This proves the linear independence of the polynomials $P_{i,I}$ in $D(X')/D(X)$ which, in view of (12), (17) finishes the proof of Theorem 2.

As pointed out in Section 2 (cf. (14)) being able to construct a basis for spaces of the type D(X) immediately allows to exhibit a basis of $D_\mu(X)$ for any $\mu \in \mathbb{C}^n$. Alternatively, using the operators $\mu_j + D_{x^j}$ instead of D_{x^j} , and (8), a basis for $D_\mu(X)$ could be constructed directly from the pieces of $T_\mu(\cdot|X)$ in the same way as shown above.

4. An example

In this section possibilities of evaluating and representing the above basis functions will briefly be discussed and illustrated by an example.

One should note first that under additional assumptions on X the above construction may simplify significantly. For instance, when X has the form $X = \{y^1,\ldots,y^1,\ldots,y^s,\ldots,y^s\}$ where $Y = \{y^1,\ldots,y^s\}$ spans $\mathbb{R}^s$, one has $|\mathcal{A}(X_0|y)| = 1$. In this case (25) produces in view of the representation (7) the expected tensor product basis. When the elements of X are in general position one has $\mathcal{A}(X|y) = \mathcal{A}(X_0|y) = \mathcal{S}$ and the different pieces of $T(\cdot|V)$, $V \subset X$, form a basis of D(X) . In this case the following explicit representation for $T(\cdot|X)$ was established by Dahmen and Micchelli (1986):

$$T(x|X) = \frac{1}{(n-s)!} \sum_{Y \in \mathcal{B}_s(X)} a_Y |\det Y|^{-1} (x^Y \cdot x)^{n-s} \chi_{<Y>_+}(x), \qquad (33)$$

where, for $Y \in \mathcal{B}_s(X)$,

$$x^Y \cdot y = 1 \ , \ y \in Y$$

and

$$a_Y = \prod_{z \notin Y} (1 + x^Y \cdot z)^{-1} \ .$$

A similar representation can be given for $T_\mu(\cdot|X)$ when $\mu_j \neq 0$, $j = 1,\dots,n$ (cf. Dahmen and Micchelli (1981)).

When X is in general position another explicit basis was constructed by Dahmen and Micchelli (1985) without using truncated powers.

In general, one has to use the recurrence relation (cf. Dahmen (1980))

$$T(x|X) = \frac{1}{n-s} \Sigma_{j=1}^n \lambda_j T(x|X \setminus \{x^j\}) \qquad (34)$$

which holds whenever

$$x = \Sigma_{j=1}^n \lambda_j x^j \ . \qquad (35)$$

A general strategy would be to select first a possibly large subset X^1 of X for which a basis is easily available either via (7) or (33). The representations for the extensions of type (25) could then be obtained by applying the recursion (34). The freedom in choosing the representations (35) should then be exploited in such a way that possibly many of the previously calculated representations occur during the recursion. This is in fact strongly favoured by the fact that (25) always involves truncated powers for subsets of X .

The following bivariate example illustrates this strategy. As pointed out before, after assigning appropriate signs to the elements of X , so that $0 \notin [X]$, X_0 will be always in general position. Hence $S = A(X_0|y)$ and $H_i = \emptyset$ (cf. (22)) for all i , which simplifies, of course, the construction.

Let $X = \{x^1,\dots,x^6\} \subset \mathbb{R}^2$ where $x^1 = x^2 = (1,0)$, $x^3 = x^4 = (0,1)$, $x^5 = (1,1)$, $x^6 = (-1,1)$. In order to construct a basis for $D(X)$ one can follow the lines of the previous section extending step by step bases of appropriate subspaces of $D(X)$. For instance, consider

$X^1 = \{x^1,\ldots,x^4\}$. As pointed out before the representation (7) already indicates the tensor product structure of $D(X^1)$ and one easily verifies that

$$D(X^1) = \mathrm{span}\{1, x_1, x_2, x_1x_2\} \quad . \tag{36}$$

Setting $X^2 = X^1 \cup \{x^5\}$, one has

$$A(X^1|x^5) = \{\{x^1\},\{x^2\},\{x^3\},\{x^4\}\} .$$

One can take $Y_1 = \{x^1\}$, $Y_2 = \{x^3\}$ as representers of the corresponding equivalence classes in $A(X^1, x^5)$, with $J_1 = J_2 = \{(0),(1)\}$. According to (25) one has

$$P_{1,(0)}(x) = T(x|X^2) \ , \ P_{1,(1)}(x) = T(x|X^2 \setminus \{x^1\})$$

for $x \in C_1 = <\{x^1,x^5\}>_+$ and

$$P_{2,(0)}(x) = T(x|X^2) \ , \ P_{2,(1)}(x) = T(x|X^2 \setminus \{x^3\})$$

for $x \in C_2 = <\{x^3,x^5\}>_+$. By (34) one obtains, for $x \in C_1$,

$$T(x|X^2) = \frac{1}{3}((x_1 - x_2)T(x|X^2 \setminus \{x^1\}) + x_2T(x|X^1)) .$$

Using (34) again yields

$$T(x|X^2 \setminus \{x^1\}) = \frac{1}{2}((x_1 - x_2)T(x|x^3,x^4,x^5) + x_2T(x|x^2,x^3,x^4)) ,$$

which for $x \in C_1$ reduces to

$$\frac{x_2}{2} T(x|x^2,x^3,x^4) = x_2^2/2 .$$

Likewise, $T(x|X^1) = x_1x_2$, so that

$$P_{1,(0)}(x) = \frac{1}{2}(x_1 x_2^2 - \frac{1}{3} x_2^3) . \tag{37}$$

Similarly, for $x \in C_1$,

$$T(x | X^2 \setminus \{x^1\}) = \frac{1}{2}((x_1 - x_2) T(x | x^3, x^4, x^5) + x_2 T(x | x^2, x^3, x^4)) = \frac{x_2^2}{2} ,$$

so that

$$P_{1,(1)}(x) = x_2^2/2 . \tag{38}$$

By symmetry one only has to interchange x_1 and x_2 to obtain

$$\begin{aligned} P_{2,(0)}(x) &= \frac{1}{2}(x_2 x_1^2 - \frac{1}{3} x_1^3) , \\ P_{2,(1)}(x) &= \frac{1}{2} x_1^2 , \end{aligned} \tag{39}$$

so that by Theorem 2 $D(X^2)$ is spanned by the polynomials

$$1 , x_1 , x_2 , x_1 x_2 , x_1^2 , x_2^2 , x_1 x_2^2 - \frac{1}{3} x_2^3 , x_2 x_1^2 - \frac{1}{3} x_1^3 .$$

Finally let

$$X^0 = X^2 \cup \{x^6\} ,$$

so that

$$A(X^2 | x^6) = \{\{x^i\} : i = 1,...,5\} .$$

Setting $Y_1 = \{x^3\}$, $Y_2 = \{x^5\}$, $Y_3 = \{x^1\}$, $J_1 = \{(0),(1)\} = J_3$, $J_2 = \{(0)\}$, one has $C_1 = <\{x^3,x^6\}>_+$, $C_2 = <\{x^5,x^6\}>_+$, $C_3 = <\{x^1,x^6\}>_+$. C_1, C_2, C_3 clearly satisfy (18). Hence, according to (25) one obtains

$$P_{1,(0)}(x) = T(x | X) , \quad P_{1,(1)}(x) = T(x | X \setminus \{x^3\})$$

for $x \in C_1$,

$$P_{2,(0)}(x) = T(x|X) \ , \ x \in C_2 \setminus C_1 \ ,$$

and

$$P_{3,(0)}(x) = T(x|X) \ , \ P_{3,(1)}(x) = T(x|X \setminus \{x^1\})$$

for $x \in C_3 \setminus C_2$.

Repeated application of (34) yields

$$P_{1,(0)}(x) = (x_1 + x_2)^4/48 \ , \ P_{1,(1)}(x) = (x_1 + x_2)^3/12 \ ,$$

$$P_{2,(0)}(x) = \frac{1}{48} x_2^2(x_1 + x_2)^2 + \frac{1}{24} x_1x_2^2(x_1 + x_2) + \frac{1}{16}(x_2 - x_1)(x_2x_1^2 - \frac{1}{3}x_1^3),$$

$$P_{3,(0)}(x) = \frac{1}{24}(x_1 + x_2)x_2^3 + \frac{1}{8}(x_1x_2^3 - \frac{1}{3} x_2^4) \ ,$$

$$P_{3,(1)}(x) = \frac{1}{6} x_2^3 \ .$$

One easily verifies (cf. (14)) that

$$\dim D(X) = |\mathcal{B}_2(X)| = 13 \ .$$

According to the preceding calculations a basis for D(X) is then given by the following thirteen polynomials

$$1 \ , x_1 \ , x_2 \ , x_1x_2 \ , x_1^2 \ , x_2^2 \ , x_1x_2^2 - \frac{1}{3} x_2^3 \ ,$$

$$x_2x_1^2 - \frac{1}{3} x_1^3 \ , x_2^3 \ , (x_1 + x_2)^3 \ , \ (x_1 + x_2)^4 \ ,$$

$$\frac{1}{48} x_2^2(x_1 + x_2)^2 + \frac{1}{24} x_1x_2^2(x_1 + x_2) + \frac{1}{16}(x_2 - x_1)(x_2x_1^2 - \frac{1}{3} x_1^3) \ ,$$

$$\frac{1}{24}(x_1 + x_2)x_2^3 + \frac{1}{8}(x_1x_2^3 - \frac{1}{3} x_2^4) \ .$$

Acknowledgement The author wishes to thank A. Ron for valuable comments and discussions that were of significant help in revising the first draft of the paper.

References

Ben-Artzi, A. and Ron, A. (1987), Translates of exponential box splines and their related spaces. Manuscript, to appear in Trans. Amer. Math. Soc.

de Boor, C. and Höllig, K. (1982/83), B-splines from parallelepipeds. J. Analyse Math., 42, 99 - 115.

Dahmen, W. (1980), On multivariate B-splines. SIAM J. Numer. Anal., 17, 179 - 191.

Dahmen, W. and Micchelli, C.A. (1981), On limits of multivariate B-splines. J. Analyse Math., 39, 256 - 278.

Dahmen, W. and Micchelli, C.A. (1983a), Translates of multivariate splines. Linear Algebra and its Applications 52/53, 217 - 235.

Dahmen, W. and Micchelli, C.A. (1983b), Recent progress in multivariate splines. In: Approximation Theory IV, ed. by C.K. Chui, L.C. Schumaker, J.D. Ward. Academic Press, 27 - 120.

Dahmen, W. and Micchelli, C.A. (1985), On the local linear independence of translates of a box spline. Studia Mathematica, 82, 243 - 263.

Dahmen, W. and Micchelli, C.A. (1988), The number of solutions to linear diophantine equations and multivariate splines. Trans. Amer. Math. Soc., 308, 509 - 532.

Dahmen, W. and Michelli, C.A. (1986), Statistical encounters with B-splines. Contemporary Mathematics, 59, 17 - 48.

Dahmen, W. and Micchelli, C.A. (1987), On multivariate E-splines. Preprint No. 267, Freie Universität Berlin, to appear in Advances in Mathematics.

Dyn, N. and Ron, A. (1988a), On multivariate polynomial interpolation. To appear in these proceedings.

Dyn, N. and Ron, A. (1988b), Local approximation by certain spaces of exponential polynomials, approximation order of exponential box splines and related interpolation problems. To appear in Trans. Amer. Math. Soc.

MONOTONE PIECEWISE CUBIC DATA FITTING

F. N. FRITSCH
Computing and Mathematics Research Division
Lawrence Livermore National Laboratory

Abstract This paper describes PCHLS, an algorithm for least squares fitting of a monotone piecewise cubic function to data. It extends the piecewise cubic Hermite interpolation package PCHIP to situations in which the data are noisy or are adequately represented by far fewer cubic pieces than data points.

Key words: Data fitting, Least squares, Data reduction, Piecewise cubic approximation, Monotonicity preserving approximation, Shape preserving approximation

1. Introduction

Necessary and sufficient conditions for a piecewise cubic function to be monotonic were published in [1]. PCHIP, a complete package for interpolation with piecewise cubic functions and evaluating the results has been in use for many years [4]. PCHLS is a new algorithm to do least squares fitting of a monotone piecewise cubic function to given univariate data (x_k, y_k), $k=1,\ldots,m$. Its development was motivated by user-expressed needs to extend PCHIP to noisy data, for which interpolation is not appropriate, or to cases in which it is clear that an adequate approximation could be achieved with far fewer cubic pieces than data points (data reduction).

In Section 2 is given the mathematical statement of the problem solved by PCHLS, and algorithmic details are described in Section 3. Several examples to illustrate the results of the algorithm appear in Section 4. Directions for further work are indicated in Section 5.

2. The Mathematical Problem

Let $[a,b]$ be an interval containing all of the data points x_k and define breakpoints t_i so that

$$a = t_1 < t_2 < \ldots < t_{nseg} < t_n = b, \tag{1}$$

where $nseg = n - 1$. A piecewise cubic Hermite (PCH) function with breakpoints t_i has the form

$$f(x) = \sum_{i=1}^{n} \left[f_i \; H_i^{\mathrm{f}}(x) + d_i \; H_i^{\mathrm{d}}(x) \right], \tag{2}$$

where $f_i{=}f(t_i)$, $d_i{=}f'(t_i)$ and $H_i^{\mathrm{f}}, H_i^{\mathrm{d}}$ are the cubic Hermite basis functions defined by

$$H_i^{\mathrm{f}}(t_j) = \delta_{ij}, \; \frac{d}{dx} H_i^{\mathrm{f}}(t_j) = 0; \tag{3}$$

$$H_i^{\mathrm{d}}(t_j) = 0, \frac{d}{dx} H_i^{\mathrm{d}}(t_j) = \delta_{ij}, \tag{4}$$

where we have used the standard Kronecker delta notation.

2.1 Least squares equations

The least squares equations for the unknown PCH parameters f_i, d_i are obtained by evaluating (2) at each of the data points:

$$\sum_{i=1}^{n} \left[f_i \; H_i^{\mathrm{f}}(x_k) + d_i \; H_i^{\mathrm{d}}(x_k) \right] = y_k, \quad k = 1, \ldots, m. \tag{5}$$

We note that if the $2n$ unknowns are arranged in pairs $f_1, d_1, f_2, d_2, \ldots, f_n, d_n$ the k-th row of the $m \times 2n$ least squares matrix will be

$$\left[H_1^{\mathrm{f}}(x_k) \; H_1^{\mathrm{d}}(x_k) \; H_2^{\mathrm{f}}(x_k) \; H_2^{\mathrm{d}}(x_k) \; \ldots \; H_n^{\mathrm{f}}(x_k) \; H_n^{\mathrm{d}}(x_k) \right]. \tag{6}$$

Since the i-th Hermite basis function is zero unless $x_k \in (t_{i-1}, t_{i+1})$, the matrix will have a block structure with (at most) four nonzero entries in any row, as indicated in Figure 1.

2.2 Monotonicity constraints

To the least squares equations (5) must be added the monotonicity constraints

$$0 \leq s_{i-1} d_i \leq 3 s_{i-1} \frac{f_i - f_{i-1}}{t_i - t_{i-1}}, \quad i = 2, \ldots, n; \tag{7}$$

$$0 \leq \; s_i d_i \; \leq 3 s_i \frac{f_{i+1} - f_i}{t_{i+1} - t_i}, \quad i = 1, \ldots, n-1. \tag{8}$$

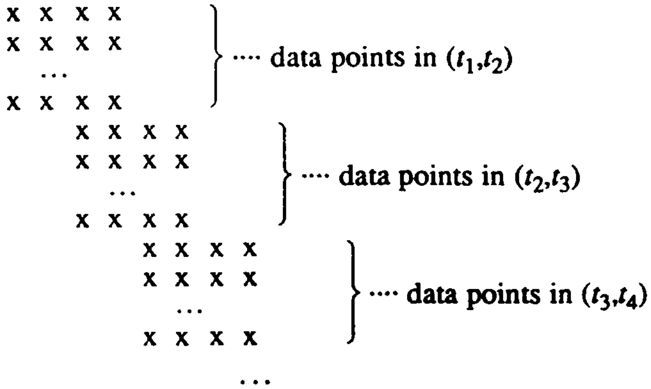

···· data points in (t_1,t_2)
···· data points in (t_2,t_3)
···· data points in (t_3,t_4)

Figure 1. Structure of least squares matrix

Here s_i will either be $+1$ or -1 according as the approximation is to be monotone increasing or decreasing in (t_i, t_{i+1}). These sufficient conditions for montotonicity come from requiring that the derivatives lie in the square $[0,3]\times[0,3]$ in the monotonicity region of [1]. Conditions (7) and (8) constitute $4(n-1)$ linear inequalities[1] among the unknowns f_i, d_i. Each involves either one or three of the unknowns and has a block structure similar to that of the least squares equations.

3. The Algorithm

PCHLS is given data (x_k, y_k) and weights w_k, breakpoints t_i satisfying (1), and an array of constraint flags s_i:

$$s_i = \begin{cases} +1 & \text{for monotone increasing in } (t_i, t_{i+1}), \\ -1 & \text{for monotone decreasing in } (t_i, t_{i+1}), \\ 0 & \text{for no constraint in } (t_i, t_{i+1}). \end{cases} \tag{9}$$

PCHLS is to return PCH coefficients (f_i, d_i), $i=1,\ldots,n$ in (2) which minimize the sum of squares of weighted residuals

$$\sum_{k=1}^{m} (w_k[f(x_k) - y_k])^2 \tag{10}$$

subject to the constraints (7) and (8).

[1] These will not all be independent. In the common case in which the approximation is to be increasing throughout $[a,b]$, for example, there will be only $n + 2(n-1)$ constraints.

The algorithm operates as follows.

Step 1.
Sort the data so that $x_1 \leq x_2 \leq \ldots \leq x_m$. (This is actually done externally to PCHLS, to avoid extra work in case the data arrives ordered.)

Step 2.
Evaluate the cubic Hermite basis functions at the data points and set up the least squares matrix. Weights are included by multiplying the k-th row (6) and the right-hand side y_k by w_k.

Step 3.
Set up the monotonicity constraints (7) and (8). No constraints are generated for intervals in which s_i=0. Redundant sign constraints are eliminated when s_{i-1}=s_i.

Step 4.
Solve this linearly constrained linear least squares problem via SLATEC subroutine LSI, which is based on the algorithm described in [5].

Step 5.
Set "small" d_i-values to zero. (This postprocessing step is necessary because the current version of LSI often returns values the size of the unit roundoff, possibly with the wrong sign, when one of the sign constraints is binding.)

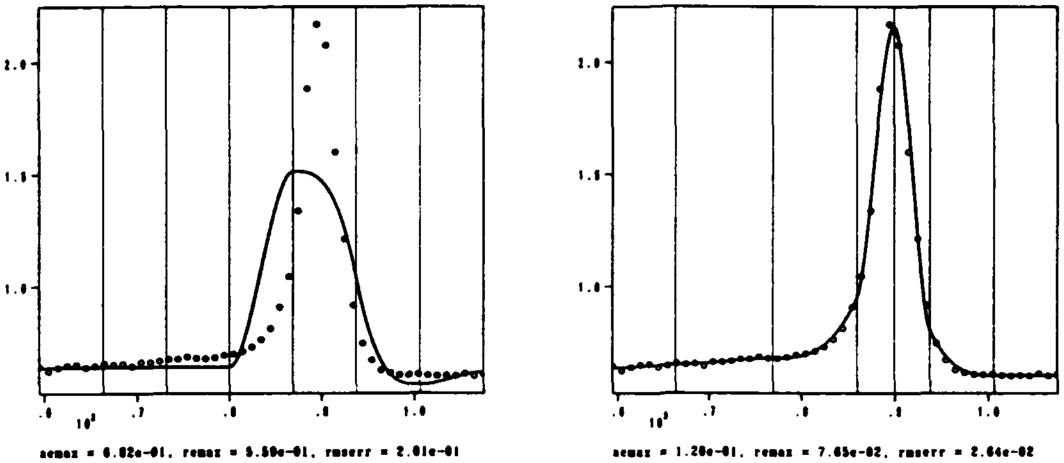
Since the hard part of obtaining a satisfactory fit is generally the choice of breakpoints and constraints, an interactive driver has been built on top of PCHLS. It reads data and weights, translating absent weights to w_k=1. After the user sets $nseg$, it generates either uniform or equidistributed[2] breakpoints and automatically selects the s_i by examining the data. It performs Step 1, above, calls PCHLS, and does Step 5. It plots the approximation, with data points and breakpoints superimposed, after each fit. The driver allows the user to move or delete breakpoints, change sign constraints, or change the number of segments.

4. Examples

This section contains some plots generated by the PCHLS driver, to illustrate the capabilities of the algorithm. All fits were unweighted.

The first example is the titanium data from [3]. This is uniformly spaced and appears to contain a moderate amount of noise. Figure 2(a) uses $nseg$=7 with all default settings and is clearly a poor fit. Adjusting the positions of breakpoints 3–5 and setting s_i=+1 in the first four intervals, -1 in the last three, yields 2(b). Changing s_4 to zero improves the RMS error from .0026 to .0018 (maximum relative error from 8% to 5%). (See Figure 3(a).) Deleting the second breakpoint has

[2] Same number of data points in each segment.



(a)

(b)

Figure 2. Two fits to titanium data. (a) The result with all default settings. (b) The result after adjusting middle breakpoints and constraints.

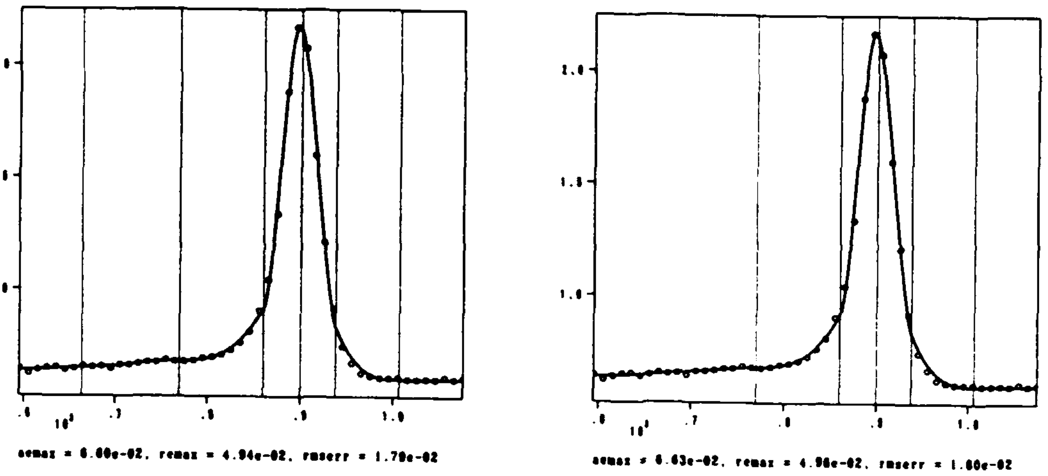
virtually no effect on the result, as indicated in Figure 3(b). In Figure 4 this curve is compared with the PCHIP interpolant. The latter has $2m$=98 parameters, whereas the PCHLS fit has only $2n$=14, so we have smoothed out the noise and achieved a data reduction factor of 7.

The second example illustrates the use of equidistributed breakpoints. The assumption here is that if the data are not uniformly spaced, then the sample rate was dictated by the behavior of the phenomonon under study. This is certainly the case with the potentiometric titration data from [6]. In figure 5(a) is the PCHLS result with seven equidistributed breakpoints. This has $2n$=14 parameters and an RMS error of 18 (maximum relative error 1%). It looks much the same as the PCHIP interpolant in Figure 5(b), which has $2m$=42 parameters.

5. Further developments

The most obvious improvement needed would be a constrained linear least squares solver that could take advantage of the sparseness of the problem. The least squares matrix is $m \times 2n$ but has only $4m$ nonzero elements. If $ncon \leq 4(n-1)$ is the actual number of constraints, there will be fewer than $3ncon$ nonzero elements in the $ncon \times 2n$ constraint matrix. While the wasted space is tolerable in most of the problems on which PCHLS has been used to date, it will be prohibitive if the algorithm is to be extended to bivariate data.

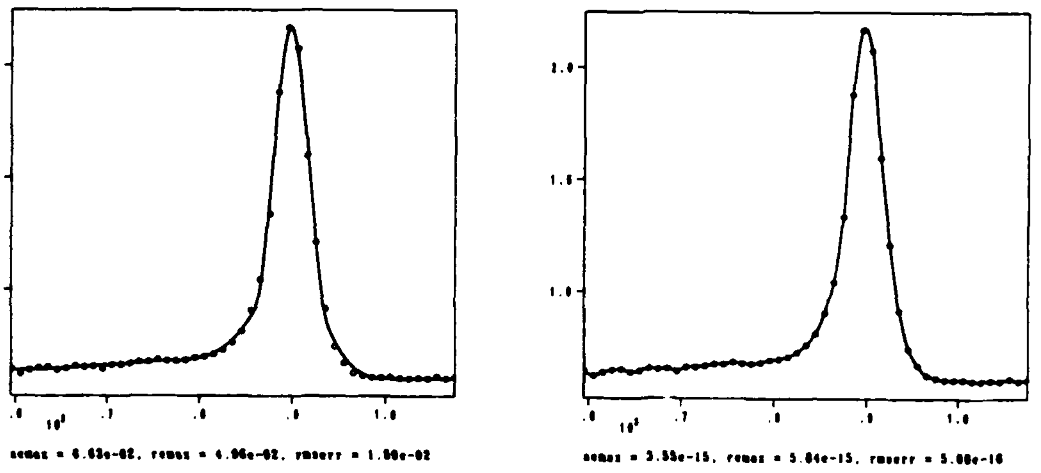
The present automatic constraint setting algorithm is very primitive. It might



(a)

(b)

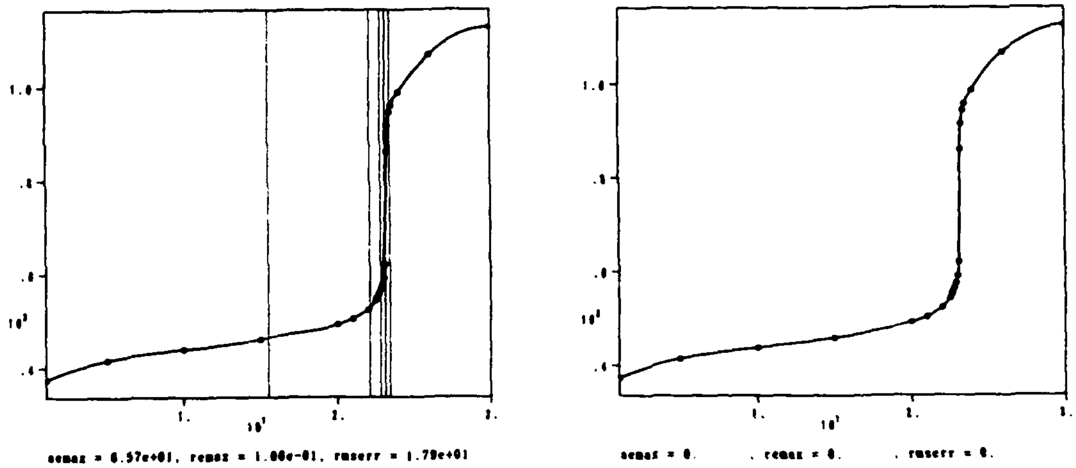
Figure 3. Two minor adjustments. (a) Removed constraint in fourth segment. (b) Deleted first interior breakpoint.



(a)

(b)

Figure 4. Comparison of the latter curve (a) with the PCHIP interpolant (b) to the titanium data.



(a)

(b)

Figure 5. Two approximations to potentiometric titration data. (a) The PCHLS fit with n=7 and equidistributed breakpoints. (b) The PCHIP interpolant.

be improved by doing local least squares fits of straight lines to the data in each breakpoint interval.

Because the most tedious part of the fitting process for a user is adjusting the locations of the breakpoints, work on automatic knot placement, such as that reported by Peter Harris at this conference [2], might be a valuable addition to the PCHLS driver.

The author has had good results on some problems by adding convexity constraints to the monotonicity constraints provided by PCHLS. This modification comes closer to providing a true shape preserving data fitting algorithm.

Another modification under consideration is the addition of a term to the objective function to reduce the size of the second derivative jumps at the data points.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore Laboratory under contract W-7405-Eng-48.

References

1. R. E. Carlson and F. N. Fritsch. Monotone piecewise cubic interpolation. *SIAM J. Numer. Anal.*, 17:238–246, 1980.

2. M. G. Cox, P. M. Harris, and Helen M. Jones. A strategy for knot placement in least squares data fitting by splines. In *(These Proceedings)*, 1988.

3. Carl de Boor. *A Practical Guide to Splines*, page 222. Springer-Verlag, New York, 1978.

4. F. N. Fritsch and R. E. Carlson. *PCHIP Final Specifications.* LLNL Computer Documentation Report UCID-30194, Lawrence Livermore National Laboratory, Livermore, California, 1982.

5. Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*, chapter 23. Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

6. Lawrence F. Shampine and Richard C. Allen, Jr. *Numerical Computing: an introduction*, page 40. Saunders, Philadelphia, 1973.

DIRECT AND CONVERSE RESULTS ON SIMULTANEOUS APPROXIMATION BY THE METHOD OF BERNSTEIN-DURRMEYER OPERATORS

M. HEILMANN, M. W. MÜLLER
Lehrstuhl Mathematik VIII
Universität Dortmund

Abstract The n-th Bernstein-Durrmeyer operator M_n results from the n-th classical Bernstein operator with weights p_{nk} if the discrete values $f(\frac{k}{n})$ in its definition are replaced by an integral over the weighted function. For integrable functions the rate of simultaneous approximation will be related to the Ditzian-Totik modulus of smoothness. For the local rate of simultaneous approximation a generalized Voronovskaja theorem is proved.

Key words: Bernstein-Durrmeyer operators, Direct results, Inverse results, Local direct results, Simultaneous approximation.

1. Introduction

The n-th Bernstein-Durrmeyer operator M_n, $n \in \mathbb{N}$, results trom the n-th classical Bernstein operator with weights $p_{nk}(x) = \binom{n}{k}x^k(1-x)^{n-k}$, $k \in \{0,...,n\}$, $x \in I = [0,1]$, if the discrete values $f(\frac{k}{n})$ in its definition are replaced by an integral over the weighted function. More precisely M_n assigns to a function $f \in L_1(I)$ the polynomial M_nf of degree n defined by

$$(M_nf)(x) = (n+1)\sum_{k=0}^{n} p_{nk}(x)\int_0^1 p_{nk}(t)f(t)dt, \qquad x \in I.$$

These operators were introduced by Durrmeyer (1967) and first studied by

Derriennic (1979). For the rate of approximation in the L_p-metric global direct and inverse theorems have been proved by Ditzian and Ivanov. Their paper contains moreover a solution of the so-called nonoptimal approximation problem. The global saturation problem has been solved by Heilmann.

In this paper we will derive

a) global direct and inverse results for the rate of weighted simultaneous approximation $\|w(M_nf-f)^{(s)}\|_p$, $s \in \mathbb{N}_0$, of functions $f \in L_p^s(I) := \{g \mid g^{(s)} \in L_p(I)\}$, $1 \leq p < \infty$ (Theorems 3, 5, 6, 7),

b) a generalized Voronovskaja theorem for the local rate of (nonweighted) simultaneous approximation $(M_nf-f)^{(s)}(x)$, $f \in L_1^s(I)$ (Theorem 8).

If $f \in L_p^s(I)$, $s \in \mathbb{N}_0$, $1 \leq p \leq \infty$, $n > s$, $x \in I$, then

$$(M_nf)^{(s)}(x) = (n+1)\alpha(n,s)\sum_{k=0}^{n-s} p_{n-s,k}(x)\int_0^1 p_{n+s,k+s}(t)f^{(s)}(t)dt, \tag{1}$$

with $\alpha(n,s) = \dfrac{(n!)^2}{(n-s)!(n+s)!} < 1$, cf. Derriennic (1979).

Occasionally we will work for convenience with operators $M_{n,s}$ given for $h \in L_p(I)$, $1 \leq p \leq \infty$, $n > s$, $x \in I$ by

$$(M_{n,s}h)(x) = (n+1)\alpha(n,s)\sum_{k=0}^{n-s} p_{n-s,k}(x)\int_0^1 p_{n+s,k+s}(t)h(t)dt\,. \tag{2}$$

Evidently $(M_nf)^{(s)} = M_{n,s}f^{(s)}$ if $f \in L_p^s(I)$. (3)

Throughout this paper C_i will denote positive constants not necessarily the same at each occurance.

2. Basic properties

For every $n \in \mathbb{N}$, $k \in \{0,\dots,n\}$, $x \in I$, there holds

$$\sum_{k=0}^{n} p_{nk}(x) = 1\,, \quad \int_0^1 p_{nk}(t)dt = \frac{1}{n+1}, \tag{4}$$

$$\frac{k}{n}p_{nk}(x) = x \cdot p_{n-1,k-1}(x)\,, \quad \int_0^1 t \cdot p_{n+s,k+s}(t)dt = \frac{k+s+1}{(n+s+1)(n+s+2)}, \tag{5}$$

$$p'_{nk}(x) = n(p_{n-1,k-1}(x) - p_{n-1,k}(x))\,, \tag{6}$$

$$\varphi(x)^2 p'_{nk}(x) = (k-nx)p_{nk}(x)\,. \tag{7}$$

(Here and in the following we put formally $p_{nk}(x) = 0$ whenever $k < 0$ or $k > n$.)

Using (4) and the Riesz-Thorin theorem (cf. Bergh, Löfström, Theorem 1.1.1) it is easily proved that

$$\|M_{n,s}h\|_p \leq C\|h\|_p,\ h \in L_p(I),\ 1 \leq p \leq \infty,\ n > s, \tag{8}$$

with a constant C that is independent of n and p.

The following lemma is useful in connection with estimates for the moments $M_{n,s}(t-x)^m(x)$, $m \in \mathbb{N}_0$.

<u>Lemma 1</u> Let $m \in \mathbb{N}_0$, $s \in \mathbb{N}_0$, $n > s$, $x \in I$ and

$$\begin{aligned} T_{n,s,m}(x) &:= \frac{n+s+1}{(n+1)\alpha(n,s)} M_{n,s}(t-x)^m(x) \\ &= (n+s+1)\sum_{k=0}^{n-s} p_{n-s,k}(x)\int_0^1 p_{n+s,k+s}(t)(t-x)^m dt. \end{aligned} \tag{9}$$

With $\varphi(x) = \sqrt{x(1-x)}$ we have the recursion formula

$$\begin{aligned} &T_{n,s,0}(x) = 1, \quad T_{n,s,1}(x) = \frac{(s+1)(1-2x)}{n+s+2}, \\ &T_{n,s,m+1}(x) \\ &= \frac{1}{n+s+m+2}\Big[\varphi(x)^2(T'_{n,s,m}(x)+2m\cdot T_{n,s,m-1}(x))+(r+m+1)(1-2x)T_{n,s,m}(x)\Big], \end{aligned} \tag{10}$$

$m \in \mathbb{N}$, and the representations

$$\begin{aligned} T_{n,s,2m}(x) &= \sum_{i=0}^{m} \beta_{i,2m}(n)\Big[\frac{\varphi(x)^2}{n}\Big]^{m-i}\cdot n^{-2i}, \\ T_{n,s,2m+1}(x) &= (1-2x)\sum_{i=0}^{m} \beta_{i,2m+1}(n)\Big[\frac{\varphi(x)^2}{n}\Big]^{m-i}\cdot n^{-2i-1}, \end{aligned} \tag{11}$$

where the $\beta_{i,2m}(n)$ and $\beta_{i,2m+1}(n)$ are independent of x and for fixed m uniformly bounded in n.

<u>Proof</u> $T_{n,s,0}(x)$ and $T_{n,s,1}(x)$ can be calculated directly. The recursion formula follows by an argument which is similar to Derriennic (1979), Proposition II.3. Then the representations (11) can easily be derived from (10) by induction. □

In section 3 the following corollary will be needed.

<u>Corollary 2</u> For every $m \in \mathbb{N}_0$, $n > s$, $n \geq 3$ and $x \in I$ we have

$$\begin{aligned} |M_{n,s}(t-x)^{2m}(x)| &\leq Cn^{-m}(\varphi(x)^2+n^{-1})^m, \\ |M_{n,s}(t-x)^{2m+1}(x)| &\leq Cn^{-m-1}(\varphi(x)^2+n^{-1})^m, \end{aligned} \tag{12}$$

where C is independent of n.

<u>Proof</u> For $x \in [0,\frac{1}{n}] \cup [1-\frac{1}{n},1]$ we have $\varphi(x)^2 \leq \frac{n-1}{n^2}$. Thus by (11)

$$|M_{n,s}(t-x)^{2m}(x)| \leq \frac{(n+1)\alpha(n,s)}{n+s+1} \sum_{i=0}^{m} |\beta_{i,2m}(n)|(\frac{n-1}{n^3})^{m-i} \cdot n^{-2i} \leq Cn^{-2m}.$$

For $x \in [\frac{1}{n},1-\frac{1}{n}]$ we have $[n\varphi(x)^2]^{-1} \leq \frac{n}{n-1} \leq \frac{3}{2}$. Thus by (11)

$$|M_{n,r}(t-x)^{2m}(x)| \leq \frac{(n+1)\alpha(n,s)}{n+s+1} n^{-m} \varphi(x)^{2m} \sum_{i=0}^{m} |\beta_{i,2m}(n)| [n\varphi(x)^2]^{-i}$$
$$\leq Cn^{-m}\varphi(x)^{2m}.$$

The second estimate of (12) is proved analogously. □

For the two monomials e_0 and e_1 we obtain by direct computation

$$(M_{n,s}e_0)(x) = 1+O(n^{-1}), \quad (M_{n,s}e_1)(x) = x+O(n^{-1}) \tag{13}$$

uniformly for $x \in I$ and $n \to \infty$.

In our first global direct theorem the rate of (nonweighted) simultaneous approximation $\|(M_n f-f)^{(s)}\|_p$ will be estimated using the second order Ditzian-Totik modulus of smoothness. In our case the r-th order modulus of smoothness ($r \in \mathbb{N}$) is given by

$$\omega_\varphi^r(f,t)_p = \sup_{0<h\leq t} \|\Delta_{h\varphi}^r f\|_p, \; f \in L_p(I), \; 1 \leq p \leq \infty, \; \varphi(x) = \sqrt{x(1-x)}, \text{ where}$$

$$\Delta_{h\varphi}^r f(x) \underset{h\varphi}{=} \sum_{k=0}^{r} (-1)^k \binom{r}{k} f(x+(\tfrac{r}{2}-k)h\varphi(x)), \text{ whenever } [x-\tfrac{r}{2}h\varphi(x), x+\tfrac{r}{2}h\varphi(x)] \subset I$$

and $\Delta_{h\varphi}^r f(x) = 0$ otherwise.

Ditzian and Totik proved in Chapter 3 the equivalence of this modulus with the modified K-functional

$$\overline{K}_\varphi^r(f,t^r)_p = \inf\{\|f-g\|_p + t^r\|\varphi^2 g^{(r)}\|_p + t^{2r}\|g^{(r)}\|_p \,;\, g \in L_p^r(I)\}.$$

3. Global results

<u>Theorem 3</u> Let $\varphi(x) = \sqrt{x(1-x)}$, $n > s$. If $f \in L_p^s(I)$, $1 \leq p < \infty$, then

$$\|(M_n f-f)^{(s)}\|_p \leq C\{\omega_\varphi^2(f^{(s)},n^{-1/2})_p + n^{-1}\|f^{(s)}\|_p\} \tag{14}$$

with a constant C independent of n.

Remarks (14) implies that $\lim_{n\to\infty} \|(M_n f-f)^{(s)}\|_p = 0$ for every $f \in L_p^s(I)$, $1 \le p < \infty$, since $\lim_{t\to 0} \omega_\varphi^2(f^{(s)},t)_p = 0$ holds true for every function of this class.

For $f \in L_p^{s+2}(I)$, $1 \le p < \infty$, (14) implies that the rate of simultaneous approximation is $\|(M_n f-f)^{(s)}\|_p = O(n^{-1})$ $(n\to\infty)$ because in this case $\omega_\varphi^2(f^{(s)},n^{-1/2})_p = O(n^{-1})$.

For the proof of Theorem 3 we still need the following lemma which can be proved in a similar way as Lemma 3.2 in Heilmann and Müller (1989).

Lemma 4 Let $s \in \mathbb{N}_o$, $n \in \mathbb{N}$, $n > s$, $u \in I$ and

$$H_n(u) = (n+1)\{\int_0^1\int_0^u - \int_0^u\int_0^1\}(u-t) \sum_{k=0}^{n-s} p_{n-s,k}(x)p_{n+s,k+s}(t)dtdx.$$

Then we have with $\varphi(u) = \sqrt{u(1-u)}$ the estimate $H_n(u) \le Cn^{-1}\varphi(u)^2$, where C denotes a constant independent of n and u.

Proof of Theorem 3 In order to prove (14) we make use of the equivalence

$$\omega_\varphi^2(f^{(s)},n^{-1/2})_p \sim \overline{K}_\varphi^2(f^{(s)},n^{-1})_p.$$

Using (3) and (8) we have for every $g \in L_p^2(I)$

$$\|(M_n f-f)^{(s)}\|_p \le C_1\|f^{(s)}-g\|_p+\|M_{n,s}g-g\|_p. \tag{15}$$

We expand g by the Taylor formula with integral remainder

$$g(t) = g(x)+(t-x)g'(x)+R_2(g,t,x), \quad R_2(g,t,x) = \int_x^t (t-u)g''(u)du.$$

(13) and (12) imply

$$\begin{aligned}\|M_{n,s}g-g\|_p &\le C_2n^{-1}[\|g\|_p+\|g'\|_p]+\|M_{n,s}R_2(g,\cdot,x)(x)\|_p\\ &\le C_3n^{-1}[\|g\|_p+\|\varphi^2g''\|_p]+\|M_{n,s}R_2(g,\cdot,x)(x)\|_p,\end{aligned} \tag{16}$$

where the last inequality results from the estimate (a) in the proof of Theorem 9.5.3 by Ditzian and Totik.

Next it will be shown that

$$\|M_{n,s}R_2(g,\cdot,x)(x)\|_p \le C_4n^{-1}\|(\varphi^2+n^{-1})g''\|_p. \tag{17}$$

In fact it is enough to prove (17) for $p = 1$ and $p = \infty$. Then the cases $1 < p < \infty$ follow by the Riesz-Thorin theorem. Utilizing (12) with $m = 1$ the proof of (17) for

$p = \infty$ is the same as for (5.10) in the paper of Ditzian and Ivanov. For <u>$p = 1$</u> (17) is derived by applying Fubini's theorem twice, then the definition of $H_n(u)$ and Lemma 4. This gives

$$\int_0^1 |M_{n,s}(R_2(g,\cdot,x)(x)|dx$$
$$\leq (n+1)\alpha(n,s)\int_0^1 \sum_{k=0}^{n-s} p_{n-s,k}(x)\int_0^1 p_{n+s,k+s}(t)\,|\int_x^t (t-u)g''(u)du\,|dtdx$$
$$= (n+1)\alpha(n,s)\int_0^1 |g''(u)|\,\{\int_0^1\int_0^u - \int_0^u\int_0^1\}(u-t)\sum_{k=0}^{n-s} p_{n-s,k}(x)p_{n+s,k+s}(t)dtdxdu$$
$$\leq \alpha(n,s)\int_0^1 |g''(u)|H_n(u)du \leq C_5 n^{-1}\|\varphi^2 g''\|_1 \leq C_5 n^{-1}\|(\varphi^2+n^{-1})g''\|_1,$$

where C_5 is independent of n. Thus (17) is proved. Together with (15) and (16) this gives

$$\|(M_n f-f)^{(s)}\|_p$$
$$\leq C_1\|f^{(s)}-g\|_p + C_3 n^{-1}[\|f^{(s)}-g\|_p + \|f^{(s)}\|_p + \|\varphi^2 g''\|_p + \|(\varphi^2+n^{-1})g''\|_p]$$
$$\leq C\{\|f^{(s)}-g\|_p + n^{-1}\|\varphi^2 g''\|_p + n^{-2}\|g''\|_p + n^{-1}\|f^{(s)}\|_p\}.$$

Taking the infimum over all g on the right hand side leads to

$$\|(M_n f-f)^{(s)}\|_p \leq C\{\overline{K}^2_\varphi(f^{(s)},n^{-1})_p + n^{-1}\|f^{(s)}\|_p\},$$

which concludes the proof of Theorem 3. □

We did not succeed to invert Theorem 3. Therefore we studied weighted approximation. Our next theorem is again a global direct theorem, but now for the rate of weighted simultaneous approximation $\|\varphi^{2s}(M_n f-f)^{(2s)}\|_p$. This rate will be estimated using the second order weighted modulus of smoothness $\omega^2_\varphi(f,t)_{\varphi^{2s},p}$ on I, whose definition is given by Ditzian and Totik, Appendix (B.1). In the proof we use the equivalence of this modulus with the following modified weighted K-functional on I

$$\overline{K}^2_\varphi(f,t^2)_{\varphi^{2s},p}$$
$$= \inf\{\|\varphi^{2s}(f-g)\|_p + t^2\|\varphi^{2s}\varphi^2 g''\|_p + t^4\|\varphi^{2s}g''\|_p\,;\, g'\in AC_{loc}(I),\ \varphi^{2s}g'' \in L_p(I)\}.$$

<u>Theorem 5</u> Let $\varphi(x) = \sqrt{x(1-x)}$, $n > 2s$. If $\varphi^{2s}f^{(2s)} \in L_p(I)$, $1 \leq p < \infty$, then
$$\|\varphi^{2s}(M_n f-f)^{(2s)}\|_p \leq C\{\omega^2_\varphi(f^{(2s)},n^{-1/2})_{\varphi^{2s},p} + n^{-1}\|\varphi^{2s}f^{(2s)}\|_p\}.$$

For the proof the representation (1) for $(M_n f)^{(2s)}$ has to be rewritten into

$$\varphi(x)^{2s}(M_n f)^{(2s)}(x) = (n+1)\sum_{k=0}^{n-2s} \beta(n,2s,k)p_{n,k+s}(x)\int_0^1 p_{n,k+s}(t)\varphi(t)^{2s}f^{(2s)}(t)dt$$

provided $\varphi^{2s}f^{(2s)} \in L_p(I)$, $1 \leq p < \infty$, $n > 2s$, where

$$\beta(n,2s,k) = \frac{[(k+s)!]^2}{k!(k+2s)!}\cdot\frac{[(n-k-s)!]^2}{(n-k)!(n-k-2s)!} < 1.$$

Again the most difficult step in the proof is to show that $\|\varphi^{2s}M_{n,2s}R_2(g,\cdot,x)\|_p \leq Cn^{-1}\|\varphi^{2s}(\varphi^2+n^{-1})g''\|_p$. For details of the proof of this and the following two theorems see our forthcoming paper , where we derive similar results for the simultaneous approximation by the general method of Baskakov-Durrmeyer operators.

If $\varphi^{2s}f^{(2s)} \in L_p(I)$ and $\omega^2_\varphi(f^{(2s)},t)_{\varphi^{2s},p} = O(t^{2(\alpha-s)})$, $0 < \alpha-s \leq 1$ (i.e. $s < \alpha \leq s+1$), which is equivalent to $\omega_\varphi^{2(s+1)}(f,t)_p = O(t^{2\alpha})$ by Ditzian and Totik, Corollary 6.3.2, then Theorem 5 implies $\|\varphi^{2s}(M_n f-f)^{(2s)}\|_p = O(n^{s-\alpha})$. For $s < \alpha < s+1$ this result can be inverted by using the Berens-Lorentz lemma.

<u>Theorem 6</u> Let $\varphi^{2s}f^{(2s)} \in L_p(I)$, $1 \leq p < \infty$, $\varphi(x) = \sqrt{x(1-x)}$, $n > 2s$, $s < \alpha < s+1$. Then $\|\varphi^{2s}(M_n f-f)^{(2s)}\|_p = O(n^{s-\alpha})$ implies $\omega_\varphi^{2(s+1)}(f,t)_p = O(t^{2\alpha})$.

Mainly as a corollary of Theorems 5 and 6 the following equivalence result is obtained.

<u>Theorem 7</u> Let $f \in L_p(I)$, $1 \leq p < \infty$, $\varphi(x) = \sqrt{x(1-x)}$, $n > 2s$, $s < \alpha < s+1$. Then the following statements are equivalent:

(i) $\omega_\varphi^{2(s+1)}(f,t)_p = O(t^{2\alpha})$

(ii) $\varphi^{2s}f^{(2s)} \in L_p(I)$ and $\|\varphi^{2s}(M_n f-f)^{(2s)}\|_p = O(n^{s-\alpha})$

Especially for $s = 0$ we obtain from Theorem 7 the equivalence $\|M_n f-f\|_p = O(n^{-\alpha}) \Leftrightarrow \omega^2_\varphi(f,t)_p = O(t^{2\alpha})$, which has been proved by Ditzian and Ivanov, Theorem 7.4.

4. Local results

Derriennic (1979) proved the local convergence (cf. Théorème II.6)

$$\lim_{n\to\infty} (M_n f)^{(s)}(x) = f^{(s)}(x),$$

provided f is integrable and bounded on I and s-times differentiable at the point $x \in I$. Our final result is a generalization of the Voronovskaja theorem proved by Derriennic (1985).

<u>Theorem 8</u> Let $f \in L_1^s(I)$, $s \in \mathbb{N}_0$, $f^{(s)}$ twice differentiable at a fixed point $x \in (0,1)$. Then $\lim_{n\to\infty} n(M_n f - f)^{(s)}(x) = \frac{d^{s+1}}{dx^{s+1}}\left[\varphi(x)^2 f'(x)\right]$, where $\varphi(x) = \sqrt{x(1-x)}$.

<u>Proof</u> Define $F(u) := \int_0^u f(t)dt$. Then $F'(u) = f^{(s)}(u)$ a.e. in I and

$$F'(x) = f^{(s)}(x),\ F''(x) = f^{(s+1)}(x),\ F'''(x) = f^{(s+2)}(x),$$

as $f^{(s)}$ is assumed to be twice differentiable in x. We consider the Taylor formula

$$F(t) = F(x)+(t-x)F'(x)+\tfrac{1}{2}(t-x)^2F''(x)+\tfrac{1}{3!}(t-x)^3F'''(x)+(t-x)^3R(t-x),$$

where $|R(t-x)| \le C$ for $t \in I$ and $\lim_{t\to x} R(t-x) = 0$.

Differentiating this formula with respect to t leads to

$$f^{(s)}(t) = f^{(s)}(x)+(t-x)f^{(s+1)}(x)+\tfrac{1}{2}(t-x)^2f^{(s+2)}(x)+\tfrac{d}{dt}[(t-x)^3R(t-x)] \quad \text{a.e. in I.}$$

Together with Lemma 1 this gives

$$\begin{aligned} &n(M_n f-f)^{(s)}(x) \qquad\qquad (18)\\ &=n\Big\{f^{(s)}(x)\Big[\frac{\alpha(n,s)(n+1)}{(n+s+1)}-1\Big]+f^{(s+1)}(x)\frac{\alpha(n,s)(n+1)(s+1)}{(n+s+1)(n+s+2)}\cdot(1-2x)\\ &\quad+\frac{1}{2}f^{(s+2)}(x)\frac{\alpha(n,s)(n+1)}{(n+s+1)(n+s+2)(n+s+3)}\\ &\quad\cdot\Big[\varphi(x)^2[2(n+1)-4(s+1)(s+2)]+(s+1)(s+2)\Big]+M_{n,s}\Big[\frac{d}{dt}[(t-x)^3R(t-x)]\Big](x)\Big\}.\end{aligned}$$

As
$$\begin{aligned}\frac{\alpha(n,s)(n+1)}{(n+s+1)}-1 &= \Big\{\prod_{j=1}^{s}(n-s+j)-\prod_{j=1}^{s}(n+1+j)\Big\}\cdot\prod_{j=1}^{s}(n+1+j)^{-1}\\ &= -s(s+1)n^{s-1}\cdot\prod_{j=1}^{s}(n+1+j)^{-1}+O(n^{-2})\end{aligned}$$

and $\lim_{n\to\infty}\frac{\alpha(n,s)(n+1)}{(n+s+1)} = 1$ we get from (18)

$$\lim_{n\to\infty} n(M_n f-f)^{(s)}(x) = -s(s+1)f^{(s)}(x)+(s+1)f^{(s+1)}(x)+\varphi(x)^2 f^{(s+2)}(x), \tag{19}$$

provided that

$$\lim_{n\to\infty} n\cdot M_{n,s}\left[\frac{d}{dt}[(t-x)^3 R(t-x)]\right](x) = 0\,. \tag{20}$$

It is obvious that (19) can be rewritten in the form of the proposition of the theorem.

Now we look at the remainder and prove (20) where we consider the case $s > 0$. For $s = 0$ we refer to the paper by Derriennic (1985). Integration by parts leads to

$$\begin{aligned} R^*(x) &:= n\cdot M_{n,s}\left[\frac{d}{dt}[(t-x)^3 R(t-x)]\right](x) \\ &= n(n+1)\alpha(n,s)\sum_{k=0}^{n-s} p_{n-s,k}(x)\int_0^1(-1)p'_{n+s,k+s}(t)\cdot(t-x)^3 R(t-x)dt\,. \end{aligned}$$

Using (6) and changing the index of summation gives

$$\begin{aligned} R^*(x) &= n(n+1)(n+s)\alpha(n,s)\left\{\sum_{k=0}^{n-s} p_{n-s,k}(x)\int_0^1 p_{n+s-1,k+s}(t)\cdot(t-x)^3 R(t-x)dt\right. \\ &\qquad \left. + \sum_{k=-1}^{n-s-1} p_{n-s,k+1}(x)\int_0^1 p_{n+s-1,k+s}(t)\cdot(t-x)^3 R(t-x)dt\right\} \\ &= n(n+1)(n+s)\alpha(n,s)\sum_{k=-1}^{n-s}(p_{n-s,k}(x)-p_{n-s,k+1}(x)) \\ &\qquad \times\int_0^1 p_{n+s-1,k+s}(t)\cdot(t-x)^3 R(t-x)dt\,, \end{aligned}$$

as we defined $p_{nk} = 0$ if $k < 0$ or $k > n$. Using again (6) and the relation (7) one obtains

$$\begin{aligned} \varphi(x)^2 R^*(x) &= \frac{n(n+1)(n+s)}{(n-s+1)}\cdot\alpha(n,s)\sum_{k=-1}^{n-s} p_{n-s+1,k+1}(x)\cdot(k+1-(n-s+1)x) \\ &\qquad \times\int_0^1 p_{n+s-1,k+s}(t)(t-x)^3 R(t-x)dt\,. \end{aligned}$$

Applying the Cauchy-Schwarz inequality gives

$$\begin{aligned} [\varphi(x)^2 R^*(x)]^2 &\le C\cdot n^4\left\{\sum_{k=-1}^{n-s} p_{n-s+1,k+1}(x)\cdot(k+1-(n-s+1)x)^2\right\} \\ &\qquad \times\sum_{k=-1}^{n-s} p_{n-s+1,k+1}(x)\left\{\int_0^1 p_{n+s-1,k+s}(t)(t-x)^3 R(t-x)dt\right\}^2 \end{aligned}$$

The term in the first curly bracket equals $(n-s+1)\varphi(x)^2$ by Lorentz's formulas given in his book, p.14. Choosing $\epsilon > 0$ there exists a $\delta > 0$ such that $|R(t-x)| < \epsilon$ whenever $|t-x| < \delta$ and we obtain

$$
\begin{aligned}
[\varphi(x)^2 R^*(x)]^2 &\leq C \cdot n^5 \varphi(x)^2 \sum_{k=0}^{n-s+1} p_{n-s+1,k}(x) \frac{1}{n+s} \Big\{ \epsilon^2 \int_0^1 p_{n+s-1,k+s-1}(t)(t-x)^6 dt \\
&\qquad + \frac{C}{\delta^2} \int_0^1 p_{n+s-1,k+s-1}(t)(t-x)^8 dt \Big\} \\
&\leq C \cdot n^3 \varphi(x)^2 \Big\{ \epsilon^2 \cdot T_{n,s-1,6}(x) + \frac{C}{\delta^2 n} \cdot T_{n,s-1,8}(x) \Big\} \\
&\leq C \varphi(x)^2 \Big\{ \epsilon^2 + \frac{C}{\delta^2 n} \Big\},
\end{aligned}
$$

where we used Lemma 1 and Corollary 2. Thus for n big enough we get

$|\varphi(x)^2 R^*(x)| \leq C\epsilon$ which gives (20) as $x \in (0,1)$. □

References

Bergh J., Löfström J., 'Interpolation spaces, An introduction', Springer-Verlag, Berlin, Heidelberg, New York, 1976.

Derriennic M. M. (1979), 'Sur l'approximation des fonctions intégrables sur [0,1] par des polynômes de Bernstein modifiés', J. Approx. Theory, 26, 277-292.

Derriennic M. M. (1985), 'Additif au papier "Sur l'approximation de fonctions intégrables sur [0,1] par des polynômes de Bernstein modifiés" ', unpublished.

Ditzian Z., Ivanov K., 'Bernstein type operators and their derivates', to appear in J. Approx. Theory.

Ditzian Z., Totik V., 'Moduli of smoothness', Springer Series in Computational Mathematics 9, Springer-Verlag, Berlin, Heidelberg, New York, 1987.

Durrmeyer J. L. (1967), 'Une formule d'inversion de la transformée de Laplace: Applications à la théorie des moments', Thèse de 3e cycle, Faculté des Sciences de l'Université de Paris.

Heilmann M., 'L_p-saturation of some modified Bernstein operators', to appear in J. Approx. Theory.

Heilmann M., Müller M. W. (1989), 'On simultaneous approximation by the method of Baskakov-Durrmeyer operators', Numer. Funct. Anal. and Optimiz., 10, 127-138.

Heilmann M., Müller M. W., 'Direct and converse results on weighted simultaneous approximation by the method of operators of Baskakov-Durrmeyer type', in preparation.

Lorentz G. G., 'Bernstein polynomials', Chelsea Publishing Company New York, N. Y., 1986.

ORTHOGONALITY AND APPROXIMATION IN A SOBOLEV SPACE

A. ISERLES
DAMTP
Univ. of Cambridge

P.E. KOCH & S.P. NØRSETT
Div. of Mathematical Sciences
Norwegian Inst. of Technology

J.M. SANZ-SERNA
Dept. of Appl. Mathematics
Univ. of Valladolid

Abstract This paper explores polynomials orthogonal with respect to the Sobolev inner product

$$(f,g)_\lambda := \int_{-1}^{1} f(x)g(x)dx + \lambda \int_{-1}^{1} f'(x)g'(x)dx, \quad \lambda \geq 0.$$

We investigate expansions in Legendre polynomials—it transpires that their coefficients satisfy an explicitly known recurrence relation. Moreover, we re-interpret a result of Althammer [1962] and Gröbner [1967] on a differential relation which is obeyed by the Sobolev-orthogonal polynomials and exploit it to derive a useful expression for the corresponding Fourier coefficients. These results lead to an efficient algorithm for approximation by polynomials in the underlying Sobolev space. This algorithm is introduced and described in detail, accompanied by numerical examples.

Key words: Fourier coefficients, Legendre polynomials, Polynomial approximation, Recurrence relations, Sobolev norm, Ultraspherical polynomials.

1. Orthogonality in a Sobolev space

The theme of the present paper is orthogonality with respect to the *Sobolev inner product*

$$(f,g)_\lambda := \int_{-1}^{1} f(x)g(x)dx + \lambda \int_{-1}^{1} f'(x)g'(x)dx, \tag{1}$$

where λ is a non-negative real parameter, while f and g range across the Sobolev space W_2^1 (the set of real functions with L_2-integrable derivatives).

Each inner product that acts on polynomials generates (e.g. by the Gram-Schmidt process) a set of *orthogonal polynomials*. Disregarding normalization for the time being, we denote them by $p_0^{(\lambda)}, p_1^{(\lambda)}, p_2^{(\lambda)}, \ldots$:

$$\int_{-1}^{1} p_m^{(\lambda)}(x)p_n^{(\lambda)}(x)dx + \lambda \int_{-1}^{1} p_m^{(\lambda)\prime}(x)p_n^{(\lambda)\prime}(x)dx \begin{cases} = 0 & : m \neq n \\ > 0 & : m = n \end{cases}. \tag{2}$$

Sobolev orthogonality has been already introduced and debated by several authors, in particular Lewis [1947], Althammer [1962] and Gröbner [1967],[1] in a more general setting: Let $\varphi_0, \varphi_1, \ldots, \varphi_L$ be $L+1$ given distributions (i.e. real, right-continuous, monotonically non-decreasing functions with an infinite number of points of increase and with all moments bounded) on the real interval $\mathcal{R}$. One may consider polynomials orthogonal with respect to the inner product

$$\int_{\mathcal{R}} f(x)g(x)d\varphi_0(x) + \int_{\mathcal{R}} f'(x)g'(x)d\varphi_1(x) + \cdots + \int_{\mathcal{R}} f^{(L)}(x)g^{(L)}(x)d\varphi_L(x). \qquad (3)$$

It is possible to prove easily that these *monic* polynomials solve the variational isoperimetric problem

$$\min\left\{\int_{\mathcal{R}} y^2(x)d\varphi_0(x) + \int_{\mathcal{R}} y'^2(x)d\varphi_1(x) + \cdots + \int_{\mathcal{R}} y^{(L)^2}(x)d\varphi_L(x)\right\},$$

where y ranges across all monic polynomials of given degree [Althammer, 1962]. Some of our present results extend to the inner product (3), and they will be subject of a forthcoming paper. In the present work we focus on the simpler (and the most useful!) form (1).

It follows readily from (2) that the underlying *Fourier coefficients* of a function f are

$$\hat{f}_n(\lambda) = \frac{(f, p_n^{(\lambda)})_\lambda}{(p_n^{(\lambda)}, p_n^{(\lambda)})_\lambda}, \ n = 0, 1, \ldots. \qquad (4)$$

Thus, if f is an m-th degree polynomial then $f \equiv \sum_{n=0}^{m} \hat{f}_n(\lambda)p_n^{(\lambda)}$, whereas for any $f \in W_2^1$ it is true that $\lim_{m\to\infty} \|f - \sum_{n=0}^{m} \hat{f}_n(\lambda)p_n^{(\lambda)}\|_\lambda = 0$. Here $\|\cdot\|_\lambda$ is the norm induced by $(\cdot, \cdot)_\lambda$ [Lewis, 1947].

A mechanism to approximate functions by polynomials in Sobolev norm is useful in numerous applications, when derivatives, and not just function values, are important, e.g. in *spectral methods* for differential equations. The standard Legendre projection (i.e. $\lambda = 0$) produces poor approximation to the derivative, which might be pointwise worse by orders of magnitude than the underlying approximation to the function: an example to this effect features in §4.

In §2 we introduce a representation of $p_n^{(\lambda)}$ as a linear combination of Legendre polynomials. The coefficients, which depend on λ, obey a known three-term recurrence relation, hence can be obtained easily.

§3 is devoted to a differential relation that is satisfied by $p_n^{(\lambda)}$. Although it has been already debated, in a different form, by both Althammer [1962] and Gröbner [1967], we provide both easier derivation and a different interpretation of the result. Our approach leads to a relationship between the Fourier coefficients (4) and the quantities

$$\hat{f}_n^* := \int_{-1}^{1} f(x)P_n^{(1,1)}(x)dx, \ n = 0, 1, \ldots$$

where $P_n^{(1,1)}$ are the $(1,1)$ *ultraspherical polynomials* [Chihara, 1978]. We produce recursively the coefficients that feature in that relationship. Results of §2 and §3 are assembled into a numerical algorithm to project functions into polynomials in W_2^1.

[1] We are grateful to Prof. W. Gautschi for drawing our attention to the above references.

Finally, in §4 we present a computational example demonstrating that, given a function with "awkward" derivative, the expansion in Legendre polynomials produces poor approximation to the derivative, while expansion in Sobolev-orthogonal polynomials brings about good approximation to both the function and its derivative.

Future papers will address themselves to the theory of Sobolev-orthogonal polynomials in a wider context.

2. Explicit representation of $p_n^{(\lambda)}$

As it calls for no extra effort, we consider polynomials orthogonal with respect to the inner product (3) with $L = 1$. We denote the polynomials orthogonal (in the conventional sense) with respect to φ_0 by $\{q_n\}$ and say that the distributions $\{\varphi_0, \varphi_1\}$ have *property* α if for every $m, n = 0, 1, \ldots$ it is true that $\int_{\mathcal{R}} q_n'(x) q_m'(x) d\varphi_1(x) = d_{\min\{m,n\}}$. It will be assumed henceforth that property α holds.

We seek $\gamma_{n,0}, \ldots, \gamma_{n,0}$ such that

$$p_n^{(\lambda)}(x) = \sum_{k=0}^{n} \gamma_{n,k}(\lambda) q_k(x). \tag{5}$$

Denoting $c_n := \int_{\mathcal{R}} q_n^2(x) d\varphi_0(x)$, $d_m := \int_{\mathcal{R}} q_n'(x) q_m'(x) d\varphi_1(x)$ (for $m \le n$) we have

$$0 = \left(p_n^{(\lambda)}, q_\ell\right)_\lambda = \gamma_{n,n} c_n + \lambda \sum_{k=0}^{n} \gamma_{n,k} d_{\min\{k,\ell\}}, \quad \ell = 0, 1, \ldots, n-1.$$

Thus, given that $C := \operatorname{diag}\{c_0, c_1, \ldots, c_n\}$, $D = \left(d_{\min\{k,\ell\}}\right)_{k,\ell=0}^{n}$ and e_n is the n-th unit vector, we have

$$(C + \lambda D)\gamma = \omega e_n, \tag{6}$$

where $\gamma_n = (\gamma_{n,0}, \gamma_{n,1}, \ldots, \gamma_{n,n})^T$ and $\omega \neq 0$ is a normalization constant, to be chosen later at our convenience. Solution of (6) follows easily by Cramer's rule:

$$p_n^{(\lambda)}(x) = \tilde{\omega} \det \begin{bmatrix} c_0 + \lambda d_0 & \lambda d_0 & \lambda d_0 & \cdots & \lambda d_0 & q_0(x) \\ \lambda d_0 & c_1 + \lambda d_1 & \lambda d_1 & \cdots & \lambda d_1 & q_1(x) \\ \lambda d_0 & \lambda d_1 & c_2 + \lambda d_2 & \cdots & \lambda d_2 & q_2(x) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda d_0 & \lambda d_1 & \lambda d_2 & \cdots & c_{n-1} + \lambda d_{n-1} & q_{n-1}(x) \\ \lambda d_0 & \lambda d_1 & \lambda d_2 & \cdots & \lambda d_{n-1} & q_n(x) \end{bmatrix}, \tag{7}$$

where $\tilde{\omega}$ is, again, a non-zero constant.

We now subtract the bottom row of (7) from the remaining rows. This, in tandem with $\gamma_{n,0} = d_0 = 0$ yields

$$p_n^{(\lambda)}(x) = \tilde{\omega} \det \begin{bmatrix} c_1 & \lambda(d_1 - d_2) & \lambda(d_1 - d_3) & \cdots & \lambda(d_1 - d_{n-1}) & q_1(x) - q_n(x) \\ 0 & c_2 & \lambda(d_2 - d_3) & \cdots & \lambda(d_2 - d_{n-1}) & q_2(x) - q_n(x) \\ 0 & 0 & c_3 & \cdots & \lambda(d_3 - d_{n-1}) & q_3(x) - q_n(x) \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & c_{n-1} & q_{n-1}(x) - q_n(x) \\ \lambda d_1 & \lambda d_2 & \lambda d_3 & \cdots & \lambda d_{n-1} & q_n(x) \end{bmatrix}.$$

Observe that, when performing *Gaussian elimination* to bring the matrix into upper triangular form, the coefficient r_ℓ used to eliminate the ℓ-th element, $\ell = 1, \ldots, n-1$, is independent of n. These coefficients satisfy

$$\lambda d_m - \lambda \sum_{k=1}^{m-1} (d_m - d_k) r_k + c_m r_m = 0, \quad m = 1, 2, \ldots, n-1. \tag{8}$$

Our contention is that the r_k's obey the three-term recurrence relation

$$\tau_\ell r_\ell = (\varsigma_\ell + v_\ell \lambda) r_{\ell-1} - \varepsilon_\ell r_{\ell-2}, \quad \ell = 3, 4, \ldots \tag{9}$$

where

$$\begin{aligned} \tau_\ell &= c_\ell (d_{\ell-1} - d_{\ell-2}); \\ \varsigma_\ell &= c_{\ell-1} (d_\ell - d_{\ell-2}); \\ v_\ell &= (d_\ell - d_{\ell-1})(d_{\ell-1} - d_{\ell-2}); \\ \varepsilon_\ell &= c_{\ell-2} (d_\ell - d_{\ell-1}); \end{aligned}$$

with the initial conditions

$$r_0 = 0; \quad r_1 = -\frac{d_1}{c_1}\lambda; \quad r_2 = -\frac{d_2}{c_2}\lambda - \frac{d_1(d_2 - d_1)}{c_1 c_2}\lambda^2.$$

Indeed, substituting (8) into (9) yields

$$\begin{aligned} &\tau_\ell r_\ell - (\varsigma_\ell + v_\ell \lambda) r_{\ell-1} + \varepsilon_\ell r_{\ell-2} \\ &= -\lambda \left\{ (d_{\ell-1} - d_{\ell-2}) \left(d_\ell - \sum_1^{\ell-1} (d_\ell - d_k) r_k \right) - (d_\ell - d_{\ell-2}) \left(d_{\ell-1} - \sum_1^{\ell-2} (d_{\ell-1} - d_k) r_k \right) \right. \\ &\quad \left. + (d_\ell - d_{\ell-1}) \left(d_{\ell-2} - \sum_1^{\ell-3} (d_{\ell-2} - d_k) r_k \right) + (d_\ell - d_{\ell-1})(d_{\ell-1} - d_{\ell-2}) r_{\ell-1} \right\} = 0 \end{aligned}$$

and, since both (8) and (9) possess unique solutions, our assertion is true.

A point of interest is that, subject to consecutive d_ℓ's being distinct and $\tau_\ell \varepsilon_\ell > 0$, (9) implies, via the Favard theorem [Chihara, 1978], that the shifted polynomials $\tilde{r}_\ell(\lambda) := -\frac{c_1}{d_1 \lambda} r_{\ell+1}(\lambda)$, $\ell = 0, 1, \ldots$ are orthogonal with respect to some real distribution.

Next, we evaluate for future use $\sum_1^{\ell-1} r_k$. The identity (8) implies that

$$\lambda \sum_{k=1}^{\ell-1} d_k r_k = -c_\ell r_\ell - d_\ell \lambda + d_\ell \lambda \sum_{k=1}^{\ell-1} r_k. \tag{10}$$

Shifting the index by one and adding $\lambda d_{\ell-1} r_{\ell-1}$ to both sides produces

$$\lambda \sum_{k=1}^{\ell-1} d_k r_k = -c_{\ell-1} r_{\ell-1} - d_{\ell-1} \lambda + d_{\ell-1} \lambda \sum_{k=1}^{\ell-1} r_k. \tag{11}$$

We now solve (10) and (11) for the unknowns $\sum_1^{\ell-1} r_k$ and $\sum_1^{\ell-1} d_k r_k$. Subject to $d_\ell \neq d_{\ell-1}$ this yields

$$\sum_{k=1}^{\ell-1} r_k = 1 + \frac{c_\ell r_\ell - c_{\ell-1} r_{\ell-1}}{\lambda (d_\ell - d_{\ell-1})} =: 1 + \sigma_\ell(\lambda). \tag{12}$$

We now proceed to evaluate the expansion of $p_n^{(\lambda)}$. To this end, we fix the normalization constant ω so that the coefficients are polynomials in λ and $p_n^{(0)} \equiv q_n$. Having eliminated the bottom row from the determinant, it follows at once that

$$p_n^{(\lambda)}(x) = q_n(x) + \sum_{\ell=1}^{n-1} r_\ell(\lambda)(q_\ell(x) - q_n(x)),$$

and (12) gives an explicit expansion of $p_n^{(\lambda)}$ in q_ℓ's,

$$p_n^{(\lambda)}(x) = -\sigma_n(\lambda)q_n(x) + \sum_{\ell=1}^{n-1} r_\ell(\lambda)q_\ell(x). \tag{13}$$

An interesting identity readily follows from the expansion: Subtract (13) from the corresponding expression for $p_{n+1}^{(\lambda)}$. This yields

$$p_{n+1}^{(\lambda)}(x) - p_n^{(\lambda)}(x) = -\sigma_{n+1}(\lambda)q_{n+1}(x) + (r_n(\lambda) + \sigma_n(\lambda))q_n(x).$$

However, since (12) implies that $\sigma_{n+1}(\lambda) = r_n(\lambda) + \sigma_n(\lambda)$, we obtain

$$p_{n+1}^{(\lambda)}(x) - p_n^{(\lambda)}(x) = -\sigma_{n+1}(\lambda)(q_{n+1}(x) - q_n(x)). \tag{14}$$

Identity (14) is useful in the explicit evaluation of the $p_n^{(\lambda)}$'s.

An example of distributions that obey property α is the Laguerre pair $d\varphi_0(x) \equiv d\varphi_1(x) = \frac{1}{\Gamma(1+\alpha)}x^\alpha e^{-x}dx$, where $0 < x < \infty$ and $\alpha > -1$. It can be proved that $d_\ell = \frac{(\alpha+2)_{\ell-1}}{(\ell-1)!}$ and that, for $\alpha = 0$, the r_ℓ's can be identified with Chebyshev polynomials of the second kind.

Legendre weights do not satisfy property α unamended. Fortunately, since, in that case, both $p_n^{(\lambda)}$ and $q_n \equiv P_n$ maintain the same parity as n, property α is "recovered" as long as attention is restricted to indices of the correct parity—the remaining coefficients vanish and are of no interest! Since, integrating by parts,

$$\int_{-1}^{1} P_m'(x)P_n'(x)dx = P_m'(1)P_n(1) - P_m'(-1)P_n(-1) - \int_{-1}^{1} P_m''(x)P_n(x)dx,$$

it follows readily that (subject to the aforementioned restriction on parity) $d_\ell = \ell(1+\ell)$. This, together with $c_\ell = \frac{2}{2\ell+1}$, (9) and (13) implies that

$$p_n^{(\lambda)} = -\sigma_n P_n + \sum_{\ell=1}^{[\frac{n-1}{2}]} r_{n-2\ell}P_{n-2\ell}, \tag{15}$$

where

$$\frac{2\ell-5}{2\ell+1}r_\ell = (2 + (2\ell-1)(2\ell-5)\lambda)r_{\ell-2} - \frac{2\ell-1}{2\ell-7}r_{\ell-4};$$

$$\sigma_n = \frac{\frac{r_n}{2n+1} - \frac{r_{n-2}}{2n-3}}{2(2n-1)\lambda};$$

with the initial conditions

$$r_0(\lambda) = 0; \quad r_1(\lambda) = -3\lambda; \quad r_2(\lambda) = -15\lambda; \quad r_3(\lambda) = -42\lambda - 105\lambda^2.$$

3. A differential equation

We restrict our attention in the present section to the Legendre distribution. Let t be an arbitrary polynomial of degree $\leq n-3$. Obviously, $\left(p_n^{(\lambda)}, (1-x^2)t\right)_\lambda = 0$. Integration by parts produces

$$\int_{-1}^{1} (1-x^2)\left(p_n^{(\lambda)}(x) - \lambda p_n^{(\lambda)\prime\prime}(x)\right) t(x)dx = 0.$$

Hence, $p_n^{(\lambda)} - \lambda p_n^{(\lambda)\prime\prime}$ is orthogonal (with respect to the distribution $(1-x^2)dx$) to all polynomials of degree $\leq n-3$ and, taking parity into account, there exist $\alpha_n(\lambda)$ and $\beta_n(\lambda)$ so that the Sobolev-orthogonal polynomial obeys the ordinary differential equation

$$p_n^{(\lambda)}(x) - \lambda \frac{d^2}{dx^2} p_n^{(\lambda)}(x) = \alpha_n(\lambda) P_n^{(1,1)}(x) + \beta_n(\lambda) P_{n-2}^{(1,1)}(x), \qquad (16)$$

where the $P_m^{(1,1)}$'s are *ultraspherical polynomials* [Chihara, 1978].

The importance of (16) becomes apparent upon the consideration of the Fourier coefficients. Integrating by parts we have

$$\begin{aligned} \hat{f}_n \|p_n^{(\lambda)}\|^2 &= \int_{-1}^{1} f(x)\left(p_n^{(\lambda)}(x) - \lambda p_n^{(\lambda)\prime\prime}(x)\right) dx + \lambda\left\{p_n^{(\lambda)\prime}(1)f(1) - p_n^{(\lambda)\prime}(-1)f(-1)\right\} \\ &= \alpha_n \hat{f}_n^* + \beta_n \hat{f}_{n-2}^* + \lambda\left\{p_n^{(\lambda)\prime}(1)f(1) - p_n^{(\lambda)\prime}(-1)f(-1)\right\}, \qquad (17) \end{aligned}$$

where $\hat{f}_m^* := \int -1^1 f(x) P_n^{(1,1)}(x)dx$, $m = 0, 1, \ldots$ Since the values of $p_n^{(\lambda)}$ and its derivative at the end-points can be easily derived from the results of §3, it remains to provide an explicit form of α_n and β_n.

Comparing coefficients of x^n in (15) and (16) yields at once

$$\alpha_n = -\frac{n+2}{2(2n+1)}\sigma_n.$$

It is an easy exercise to demonstrate that

$$\int_{-1}^{1} P_m^{(1,1)}(x)dx = \frac{2(1+(-1)^m)}{m+2}; \qquad (18)$$

$$\int_{-1}^{1} x P_m^{(1,1)}(x)dx = \frac{2(1+(-1)^{m+1})}{m+2}. \qquad (19)$$

Let n be even. Then integration of (16) from -1 to 1, in tandem with (18), yields

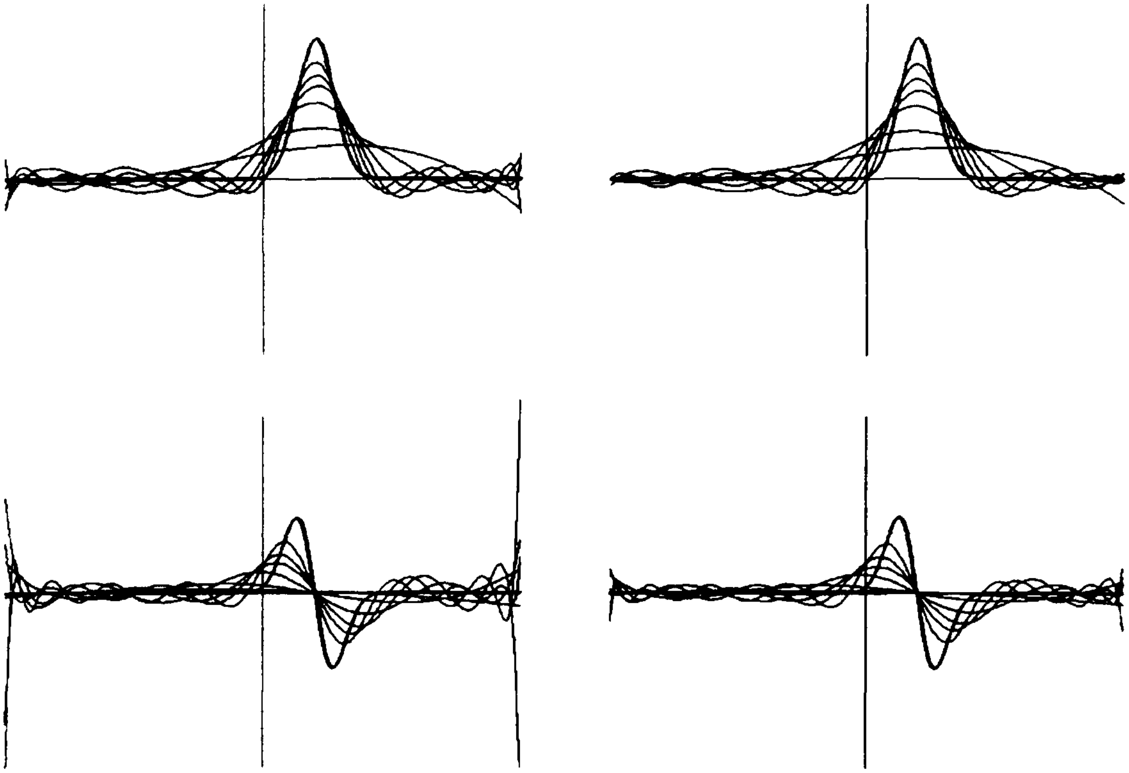
$$\frac{\alpha_n}{n+2} + \frac{\beta_n}{n} = -\frac{\lambda}{2} p_n^{(\lambda)\prime}(1).$$

Likewise, for odd n, we multiply (16) by x, integrate and employ (19): Since, by (17) and integration by parts respectively,

$$\begin{aligned} \int_{-1}^{1} x p_n^{(\lambda)}(x)dx &= \frac{2}{3}r_1; \\ \int_{-1}^{1} x p_n^{(\lambda)\prime\prime}(x)dx &= 2\left(p_n^{(\lambda)\prime}(1) - p_n^{(\lambda)}(1)\right), \end{aligned}$$

we obtain

$$\frac{\alpha_n}{n+2} + \frac{\beta_n}{n} = \frac{1}{6} r_1 - \frac{\lambda}{2}\left(p_n^{(\lambda)\prime}(1) - p_n^{(\lambda)}(1)\right).$$



Legendre projection and its derivative

Sobolev projection and its derivative

Figure 1: *Legendre and Sobolev projections.*
The thicker line denotes the function f, whereas the thin lines stand for polynomials of degrees $3, 6, \ldots, 18$.

All the ingredients for an algorithm for the evaluation of a projection of f into n-th degree polynomials in W_2^1 are now in place: either (14) or (15) are used to evaluate $p_m^{(\lambda)}$'s, the quantities $\hat{f}_m^*$, which are independent of λ, are evaluated e.g. by numerical quadrature and substituted into (17), whereas α_n and β_n are also available from explicit formulae of this section. The norms $\|p_n^{(\lambda)}\|^2$ can be evaluated either directly from the explicit form or via a recurrence relation which is omitted here due to space constraints.

The satisfaction of an "interesting" ordinary differential equation by $p_n^{(\lambda)}$ has been already noticed by Althammer [1962], who obtained, in a considerably longer derivation, the right-hand side in terms od derivatives of Legendre polynomials, and by Gröbner [1967], who identified it with the n-th derivative of a Lagrange multiplier function of the isoperimetric variational problem from §1. However, none has used it to facilitate the evaluation of Fourier coefficients. Moreover, our method yields itself to other distributions as well. The characterization of all such distributions will be described in a forthcoming paper.

4. A numerical example

The approach of this paper becomes valuable when we wish to approximate a function f by its projection into polynomials and, simultaneously, to approximate its derivative by the derivative of the polynomial approximant. Given that the derivative of f is steep, it is only to be expected that the quality of the projection in the conventional L_2 norm (the *Legendre projection*) deteriorates. Several computational examples show that in this case the *Sobolev projection* is superior. Figure 1 displays approximants of both types (with $\lambda = 0$ and $\lambda = \frac{1}{100}$ respectively) to the function

$$f(x) = e^{-100(x-\frac{1}{5})^2}, \quad -1 < x < 1,$$

by polynomials of degrees $n = 3, 6, 9, \ldots, 18$. It is evident that the Legendre projection is poor near the end-points, whereas the Sobolev projection displays reasonably good behaviour throughout the interval.

References

P. Althammer [1962], Eine Erweiterung des Orthogonalitätsbegriffes bei Polynomen und deren Anwendung auf die beste Approximation, *J. Reine Angew. Math. 211*, pp. 192–204.

T.S. Chihara [1978], *An Introduction to Orthogonal Polynomials*, Gordon & Breach, New York.

W. Gröbner [1967], Orthogonale Polynomsysteme die gleichzeitig mit $f(x)$ auch deren Ableitung $f'(x)$ approximieren, in *Funktionalanalysis, Approximationstheorie, Numerische Mathematik* (L. Collatz, G. Meinardus & H. Unger, eds.), Birkhäuser, Basel.

D.C. Lewis [1947], Polynomial least squares approximations, *Amer. J. Math. 69*, pp. 273–278.

PIECEWISE POLYNOMIAL APPROXIMATION OF POLYNOMIAL CURVES

M.A. LACHANCE
Department of Mathematics
University of Michigan - Dearborn

Abstract An algorithm based upon Chebyshev economization and constrained Chebyshev polynomials is discussed for producing low degree piecewise polynomial approximations for high degree polynomials. The method is shown to compare favorably to the Remez algorithm.

Key words Constrained Chebyshev polynomials, Chebyshev economization, Remez algorithm

1. Introduction

The majority of European and Asian computer-aided design (CAD) systems use parametric polynomials to represent 3D curves and surfaces. These systems differ in their internal representation of polynomials (power, Bernstein, and Hermite bases) and in their maximum degree (three, seven, nine, fifteen, and twenty-one). In order for these systems to exchange geometric data, representations must occasionally be altered, and/or degrees must be reduced. The numerical stability of altering representations has been recently investigated by Heybrock (1987) and Shearer

(1988). In this note we are concerned with the problem of degree reduction.

In general, a high degree polynomial will be approximated rather poorly by a lower degree polynomial, and hence piecewise or spline approximates are required. We shall describe a piecewise polynomial construction which is recursive, is uniformly near-best, controls geometric continuity, and is relatively inexpensive. The construction is compared with the Remez algorithm.

2. Constrained Chebyshev polynomials

Lachance, Saff and Varga (1979) introduced the collection of constrained Chebyshev polynomials as a solution to a minimax problem with Jacobi-type weights. These polynomials, denoted by $T_m^{(\alpha,\beta)}(x)$, are the unique monic polynomials which are extremal for

Problem 2.1 For each triple of nonnegative integers (m,α,β), determine

$$\underset{g\in\pi_{m-1}}{\text{minimum}} \quad \underset{-1\le x\le 1}{\text{maximum}} \quad (1-x)^{\alpha}(1+x)^{\beta}|x^m - g(x)|.$$

Here, π_m denotes the collection of real polynomials of degree at most m, and $\pi_{-1} = \{0\}$.

These constrained Chebyshev polynomials are a proper generalization of the classical Chebyshev polynomials (of the first kind) since $2^{m-1}T_m^{(0,0)}(x) = T_m(x)$. Most CAD systems parametrize curves on the interval $[0,1]$. We shall do the same, and introduce more convenient notation for the constrained Chebyshev polynomials on this interval.

Definition 2.1 For each pair of nonnegative integers (m,α), with $m \ge 2\alpha$, the constrained Chebyshev polynomials are defined by

$$C^{\alpha}_{m}(s) := 2^{-m}\, T^{(\alpha,\alpha)}_{m-2\alpha}(2s-1) = s^{\alpha}(s-1)^{\alpha}(s^{m-2\alpha} + \cdots);$$

$$E^{\alpha}_{m} := \max_{s\epsilon[0,1]} |C^{\alpha}_{m}(s)|.$$

Thus m denotes the degree of the polynomial and α denotes the order zero at the endpoints 0 and 1.

3. The Construction

Let $P(s) \epsilon \pi^{3}_{n}$, $s\epsilon[0,1]$, denote a vector-valued polynomial of degree n in R^3 and let E denote a prescribed tolerance. We wish to construct, within this tolerance, a visually smooth piecewise polynomial approximate to $P(s)$ of degree m. We do so applying Chebyshev economization (CE) in a component-wise manner.

Let $k = n - m$, and define the sequence of polynomials $\{P_i\}_{i=1}^{k}$ by

$$P_{i+1}(s) := P_i(s) - \frac{P_i^{(n-i)}(0)}{(n-i)!} C^{\alpha}_{n-i}(s), \quad i = 0,\ldots,k-1$$

where $P_0(s) := P(s)$. The use of the constrained Chebyshev polynomials was suggested by Lachance (1988) in order to control geometric continuity. A pointwise bound on the error between the original polynomial and the final approximate $P_k(s)$ is given by

$$\|P(s) - P_k(s)\|_E \le \sum_{i=0}^{k-1} \|P_{i+1}(s) - P_i(s)\|_E$$

$$\le \sum_{i=0}^{k-1} \left\|\frac{P_i^{(n-i)}(0)}{(n-i)!}\right\|_E E^{\alpha}_{n-i},$$

where $\|\cdot\|_E$ denotes the Euclidean norm in R^3. Clearly, if the above error exceeds the prescribed tolerance E, then the approximation is unacceptable. In this event the original curve may be partitioned into two or more pieces, the pieces reparametrized to the interval [0,1]; and then the construction may be applied to each piece (see Shearer (1987)).

Since there is no a priori indication that the final error will be satisfactory, the difficulty that arises is how one should break up the original curve. Simple bisection leaves two polynomial curves which must be approximated, again with no prior knowledge that the reduction scheme will be successful, perhaps leading to a further refinement of $P(s)$. A more efficient approach is to force the errors at each stage to fall below some threshold, say

$$\|P_{i+1}(s) - P_i(s)\|_E = \frac{\|P_i^{(n-i)}(0)\|_E}{(n-i)!} E^{\alpha}_{n-i} \le 2^{i-k}E,$$

for $0 \le i \le k-1$. With the stepwise errors so bounded, the accumulated error will always be less than E. If the stepwise error should exceed this bound at any step, then the polynomial $P_i(s)$, instead of $P(s)$, can be split and reparametrized into $P_i(s/2)$ and $P_i((1+s)/2)$, $s \epsilon [0,1]$. This latter approach involves maintaining a stack of polynomials to be economized, and has the advantage of adaptively refining the original curve.

No matter which partitioning scheme is used, the geometric continuity of the final piecewise approximate is controlled by the value of the parameter α. The approximate will be continuous if $\alpha=1$, tangentially smooth if $\alpha=2$, and curvature continuous if $\alpha=3$. This is assured by the fact that each iterate $P_i(s)$ matches tangent and acceleration directions where joined, depending on the value α. That is,

$$P_i^{(j)}(\delta) = P^{(j)}(\delta),\ j = 0,\ldots,\alpha-1;\ \delta = 0,1.$$

For a complete discussion of geometric versus parametric continuity we refer the interested reader to Bartels, Beatty and Barsky (1987).

4. Chebyshev Economization and the Remez Algorithm

At the heart of the preceding section is constrained Chebyshev economization. It is natural to ask how this construction compares with existing methods, and can it be improved upon. In this section we make some qualitative observations about this approximation process, and compare this method to an implementation of the Remez algorithm.

4.1 Chebyshev Economization

The Chebyshev expansion for a function on an interval has long been a tool of numerical analysts. The truncation of that series approximates the function almost uniformly, with an error on the order of the first neglected term. Thus it is not difficult to see that if an n^{th} degree polynomial is approximated using CE by a constant polynomial, the coefficients in the degree reduction process are, in effect, the coefficients of the polynomial's Chebyshev expansion. Consequently, terminating the process at an intermediate stage produces a near uniform approximation to the original polynomial.

The cost in arithmetic operations to produce an m^{th} degree polynomial approximate to an n^{th} polynomial is (n^2-m^2+n-m). If a breakup of the original curve is required, then this cost times the number of pieces provides an upper bound on the total number of operations. This cost can, of course, be reduced if the original curve is refined adaptively, as suggested in the previous section.

4.2 The Remez Algorithm

The Remez algorithm is a standard method for estimating the best uniform polynomial approximate of a continuous function. It exploits the fact that the error between a function and its best m^{th} degree polynomial approximate attains its maximum (in absolute value) at least $m+2$ times, and does so while alternating signs. We will not discuss the method completely here, but only summarize some of the computational steps involved. Details can be found in Davis (1963).

Let $f(s)$ denote a real function to be approximated on an interval $[a,b]$, and let $\{s_i\}_{i=1}^{m+1}$ be an initial alternation set. A polynomial $p(s) \epsilon \pi_m$ and an error E are determined so that $f(s_i)-p(s_i) = (-1)^i E$, $1 \le i \le m+2$. The location of a maximum of $|f(s)-p(s)|$, say s^*, is determined and exchanged with an appropriate element of $\{s_i\}_{i=1}^{m+1}$. The entire process is repeated with the new alternation set until the actual maximum is satisfactorily close to the computed value E.

Each iteration of the Remez algorithm requires a number of calculations. If $f(s) \epsilon \pi_n$, then the initialization of the $m+2$ dimensional system of equations requires $(m+2)(2n+m(m-1)/2)$ operations, the solution of this system of equations requires on the order of $2(m+2)^3/3$ operations, and the difficulty of estimating the maxima of $|f(s)-p(s)|$ depends on the relative precision required. Some computational savings can be derived from the fact that the coefficient matrix need not be entirely redefined at each stage, but solution of the resulting system and the search for maxima can be costly.

To more directly compare CE with the Remez algorithm, consider an example where $n=15$ and $m=7$. A single pass through the CE process takes 184 arithmetic operations. The initialization of the system of equations for the Remez algorithm requires 459 arithmetic operations, to say nothing of the solution of the 9x9 system, or the search for the maxima.

5. Zolotareff Economization

Examination of the CE algorithm shows that it is a rather simple process involving only the coefficients of a set of fixed real polynomials, and that the degree of the polynomial is reduced by one in each iteration. It seems reasonable to seek a greater degree reduction with comparable arithmetic simplicity.

To investigate this possibility consider the real valued polynomial $p(s) = as^n + a\sigma s^{n-1} + \cdots \in \pi_n.$ Conventional CE would have us define the first two approximates $p_1(s)$ and $p_2(s)$ by

$$p_1(s) = p(s) - a\, C_n^0(s) \in \pi_{n-1};$$

$$p_2(s) = p_1(s) - a(\sigma + n/2)\, C_{n-1}^0(s) \in \pi_{n-2}$$

Alternaltively, we could define an approximate $q(s)$ by $q_2(s) = p(s) - a\, Z_n(\sigma;s) \in \pi_{n-2}$, where

$$Z_n(\sigma;s) = s^n + \sigma s^{n-1} + \cdots \in \pi_n;$$

$$\max_{s\in[0,1]} |Z_n(\sigma;s)| = \min_{g\in\pi_{n-2}} \max_{s\in[0,1]} |s^n + \sigma s^{n-1} - g(s)|.$$

The extremal polynomials $Z_n(\sigma;s)$ were studied by Chebyshev's student Zolotareff (but with the problem phrased on the interval $[-1,1]$).

Zolotareff showed that $Z_n(\sigma;s)$ could be explicitly determined for only a small interval of σ values; and Kirchberger developed a first order approximation to $Z_n(\sigma;s)$ for values outside that interval (see Meinardus

(1967)). In the explicit case the extremal polynomial can be determined by a translation of the Chebyshev polynomials. In effect,

$$Z_n(\sigma;s) = s^n + \sigma s^{n-1} + \sum_{i=0}^{n-2} a_i(\sigma)\ s^i,$$

where the coefficients $a_i(\sigma)$ are polynomials in the variable σ. Otherwise $Z_n(\sigma;s)$ can be approximated by a linear combination of $T_n(s)$ and $U_n(s)$, the Chebyshev polynomials of the first and second kind, respectively.

The arithmetic complexity of using either of these representations in an economization scheme is at best no better than two iterates of conventional economization. As a result it would seem that CE is in some sense optimal relative to the quality of the result and the arithmetic computations required.

References

Bartels, R.H., Beatty, J.C., and Barsky, B.A. (1987), 'An Introduction to Splines for use in Computer Graphics and Geometric Modeling', Morgan Kaufmann Publishers, Los Altos.

Davis, P.J. (1963), 'Interpolation and Approximation', Dover Publications, New York.

Heybrock, P.C. (1987), 'Algorithms for conversion between alternative representations of parametric curves and surfaces', M.Sc. Thesis, Department of Applied Computing and Mathematics, Cranfield Institute of Technology.

Lachance, M.A. (1988), Chebyshev economization for parametric surfaces, Computer Aided Geometric Design 5, Elsevier Science Publishers, pp. 195-208.

Lachance, M.A., Saff, E.B., and Varga, R.S. (1979), Bounds on incomplete polynomials vanishing at both end points of an interval, in: 'Constructive Approaches to Mathematical Models', Academic Press, New York, 421-437.

Meinardus, G. (1967), 'Approximation of Functions: Theory and Numerical Methods', Springer Verlag, New York.

Shearer, P.A. (1987), 'Chebyshev economization for parametric curves and surfaces', M.Sc. Thesis, Department of Applied Computing and Mathematics, Cranfield Institute of Technology.

Shearer, P.A. (1988), Conversion between different mathematical representations of the polynomial types of parametric curves and surfaces, a Report prepared for the Esprit CAD*I Project, WG3.CIT.0003.88.

CALCULATION OF THE ENERGY OF A PIECEWISE POLYNOMIAL SURFACE

EWALD QUAK

LARRY L. SCHUMAKER

Department of Mathematics

Vanderbilt University

Abstract: This paper is related to the problem of constructing a smooth piecewise polynomial surface fit to data given at the vertices of a triangulation. We are interested in methods in which the surface is represented in Bernstein-Bézier form, and the coefficients are chosen so that the resulting surface minimizes some expression involving the energy of the surface (for example, see [1,3,5-9]). Here we do not go into the details of such methods themselves, but instead concentrate on the problem of representing the energy associated with a particular choice of coefficients. The formulae developed here will be applied elsewhere to specific interpolation and smoothing methods.

Keywords: multivariate splines, data fitting, interpolation, energy

1. Introduction

Let Δ be an arbitrary triangulation of a domain $\Omega \subset \mathbb{R}^2$. Given integers $0 \leq r < d$, we define the associated *space of polynomial splines of degree d and smoothness r* by

$$S_d^r(\Delta) = \{s \in C^r(\Omega) : s|_{T_i} \in \mathcal{P}_d, \quad i = 1, \ldots, N\}, \tag{1}$$

where $\mathcal{P}_d$ is the $\binom{d+2}{2}$ dimensional linear space of bivariate polynomials of total degree d, and where $T_i, i = 1, \ldots, N$, are the triangles of Δ. While the constructive theory of these spaces of splines is not yet complete, they are clearly useful tools for fitting surfaces to data of the form

$$z_i = f(x_i, y_i) + \varepsilon_i, \quad i = 1, \ldots, V, \tag{2}$$

where $(x_i, y_i), i = 1, \dots, V$, are given points scattered in the plane (and forming the vertices of the triangulation), and where $\varepsilon_i, i = 1, \dots, V$, are measurement errors.

In this paper we are interested in two classes of surface fitting methods using the space $S_d^r(\Delta)$. The first class of methods involves minimizing some measure of smoothness $J(u)$ over the subset

$$U = \{u \in S_d^r(\Delta) \ : \ u(x_i, y_i) = z_i, \quad i = 1, \dots, V\}. \tag{3}$$

The set U is the set of *interpolating splines*. We may call these kinds of methods *smoothest interpolation methods.*

The second class of methods of interest involve minimizing a combination of smoothness and goodness of fit such as

$$\rho_\lambda(s) = \lambda J(s) + E(s),$$

over $S_d^r(\Delta)$, where $\lambda \geq 0$ is a smoothing parameter, and where $E(s)$ measures the goodness of fit. For example, we might take

$$E(s) = \sum_{i=1}^{V} [s(x_i, y_i) - z_i]^2 .$$

This is an example of a *penalized least squares method* (see [6] in this volume).

While other energy expressions can be treated by similar methods, throughout the remainder of this paper we restrict our attention to the functional

$$J(u) = \sum_{i=1}^{N} J_{T_i}(u), \tag{4}$$

where

$$J_{T_i}(u) = \int\int_{T_i} [(u_{xx})^2 + 2(u_{xy})^2 + (u_{yy})^2] dxdy, \quad i = 1, \dots, N,$$

where u_{xx} represents the second order derivative of u in the x direction, etc. This expression represents the energy of a "thin plate", and is a natural way of measuring the roughness of a surface (see [3,9]). We have defined J in (4) as a sum of integrals over the individual triangles of Δ in order to be able to apply it to spline spaces $S_d^r(\Delta)$ even if the global smoothness r is less than 2. (Indeed, in applications it is common to use C^1 splines, where $r = 1$).

The main purpose of this paper is to find formulae for representing $J(u)$ in terms of the coefficients of the Bernstein-Bézier representation of a spline u in $S_d^r(\Delta)$. This is an essential step in deriving numerical algorithms for calculating spline fits as discussed above. The remainder of this paper is organized as follows. In Section 2 we introduce the necessary notation for representing splines in $S_d^r(\Delta)$ in Bernstein-Bézier form, and introduce certain energy expressions. In Section 3 we derive convenient recursion formulae for these energy expressions. In Section 4 we apply our results to give explicit formulae for cubic and quartic splines.

2. The Bernstein-Bézier representation

In working with $S_d^r(\Delta)$ numerically, it will be useful to regard $S_d^r(\Delta)$ as the linear subspace of $S_d^0(\Delta)$ defined by

$$S_d^r(\Delta) = \{s \in S_d^0(\Delta) \ : \ s \in C^r(\Omega)\}. \tag{5}$$

Each spline $s \in S_d^0(\Delta)$ can be written in the form

$$s(x,y) = s_l(x,y) \quad \text{for} \quad (x,y) \in T_l, \quad l = 1,\ldots,N,$$

where s_l is a polynomial of degree d, $l = 1,\ldots,N$. Each of these polynomials can be written in Bernstein-Bézier form as

$$s_l(r,s,t) = \sum_{i+j+k=d} c_{ijk}^l \frac{d! r^i s^j t^k}{i!j!k!},$$

where (r,s,t) are the barycentric coordinates of a point (x,y) in the triangle T_l. Identifying the *Bézier-ordinates* c_{ijk}^l on common edges of triangles forces the continuity of the associated piecewise polynomial function.

With each Bézier–ordinate c_{ijk}^l we associate a *domain point*

$$P_{ijk}^\ell = \left(iV_1^\ell + jV_2^\ell + kV_3^\ell\right)/d,$$

where V_1^ℓ, V_2^ℓ, and V_3^ℓ denote the vertices of the triangle T_l. We omit the superscript l whenever this will cause no confusion. The set of all domain points is denoted by $\mathcal{B}_d(\Delta)$. On each triangle T of Δ there are precisely $(d+1)(d+2)/2$ points of $\mathcal{B}_d(\Delta)$ spaced uniformly over T. It is common to associate the coefficient c_{ijk}^ℓ with the domain point P_{ijk}^ℓ. Clearly, the set

$$\mathcal{C} = \{c_{ijk}^\ell\}, \quad \ell = 1,\ldots,N \quad \text{and} \quad i+j+k = d$$

of coefficients uniquely defines a spline in $S_d^0(\Delta)$. It is easy to see that

$$nc = \#(\mathcal{C}) = V + (d-1)E + \binom{d-1}{2}N,$$

where V is the number of vertices in Δ (which is also the number of data points), and where E is the number of edges and N the number of triangles of Δ.

It is well-known (cf. [2,4]) that a spline $s \in S_d^0(\Delta)$ lies in $S_d^r(\Delta)$ if and only if an appropriate set of continuity conditions, each of which can be described by a linear equation involving the Bernstein-Bézier coefficients, is satisfied. Thus, we can write

$$S_d^r(\Delta) = \{s \in S_d^0(\Delta) \ : \ Ac = b\}, \tag{6}$$

where c is the coefficient vector of length nc, A is an appropriate $m \times nc$ matrix (where m denotes the number of continuity conditions enforced), and b is an appropriate m vector. The exact nature of A and b is of no concern here, but is, of course, critical in implementing a specific surface fitting method. We remark that in general, $nc > V + m$, so that even after specifying both smoothness and interpolation conditions, there remain a certain number of free parameters to be used in the minimization process.

3. Formulae for the energy of a single patch

In this section we give formulae for the energy of a polynomial of degree d described in Bernstein-Bézier form on a triangle T. Throughout this section we suppose that the polynomial is written in the form

$$p(r,s,t) := \sum_{i=0}^{d} \sum_{j=0}^{i} b_{\frac{i(i+1)}{2}+j+1} \Phi_{d-i,i-j,j}(r,s,t), \tag{7}$$

where, in general

$$\Phi_{\alpha,\beta,\gamma}(r,s,t) = \frac{(\alpha+\beta+\gamma)! r^{\alpha} s^{\beta} t^{\gamma}}{\alpha!\,\beta!\,\gamma!}. \tag{8}$$

Now since $p \in \mathcal{P}_d$, it follows that p_{xx}, p_{xy}, and p_{yy} belong to $\mathcal{P}_{d-2}$ and that $(D_{xx}^2 + 2D_{xy}^2 + D_{yy}^2)p$ belongs to $\mathcal{P}_{2(d-2)}$. Introducing the notation $\mathbf{b} := (b_1, ..., b_n)^T$, where $n = \binom{d+2}{2}$, we find that the energy of p over the triangle T is given by the quadratic form

$$J_T(p) = \mathbf{b}^T E^{(d)} \mathbf{b},$$

where $E^{(d)}$ is an appropriate symmetric $n \times n$ matrix.

The remainder of this section will be devoted to finding the entries of $E^{(d)}$. We determine them recursively, starting with the case $d = 2$. As we are dealing with a rotation invariant energy expression, without loss of generality we can assume that the triangle T is in the canonical position shown in Figure 1 with vertices $V_1 = (b,c)$, $V_2 = (0,0)$, and $V_3 = (a,0)$. In this case, the barycentric coordinates (r,s,t) of a point (x,y) are given by the formulae $r = y/c$, $s = 1 - x/a + (b-a)y/ac$, and $t = x/a - by/ac$.

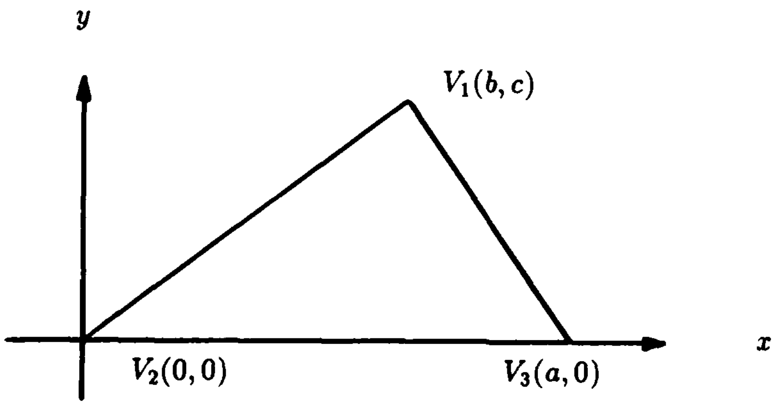


Figure 1. The canonical triangle

Now

$$p_{rr}(r,s,t) = d(d-1)\sum_{i=0}^{d-2}\sum_{j=0}^{i} b_{\frac{i(i+1)}{2}+j+1}\Phi_{d-2-i,i-j,j}(r,s,t),$$

with similar expressions for the other second-order partial derivatives. Using

$$p_{xx} = p_{rr}r_x^2 + 2p_{rs}r_xs_x + 2p_{rt}r_xt_x + p_{ss}s_x^2 + 2p_{st}s_xt_x + p_{tt}t_x^2 \tag{9}$$

and the analogous expressions for p_{xy} and p_{yy}, we find that $E^{(2)}/2ac$ is a symmetric 6×6 matrix whose entries in the upper triangular part are as follows:

$$\begin{pmatrix} (RR)^2 & 2RR\,RS & 2RR\,RT & (RS)^2 & 2RS\,RT & (RT)^2 \\ & 2RR\,SS + 2(RS)^2 & 2RR\,ST + 2RS\,RT & 2RS\,SS & 2SS\,RT + 2RS\,ST & 2RT\,ST \\ & & 2RR\,TT + 2(RT)^2 & 2RS\,ST & 2TT\,RS + 2RT\,ST & 2RT\,TT \\ & & & (SS)^2 & 2ST\,SS & (ST)^2 \\ & & & & 2SS\,TT + 2(ST)^2 & 2ST\,TT \\ & & & & & (TT)^2 \end{pmatrix}$$

where $R = (r_x, r_y)$, $S = (s_x, s_y)$, and $T = (t_x, t_y)$, each of which can be expressed in terms of the quantities a,b and c describing the locations of the vertices of T.

We now present the main result of this paper, a recursion relation for computing the entries in the matrix $E^{(d)}$ for arbitrary $d > 2$.

Theorem 1. *Let T be a canonical triangle as in Figure 1. Then for $i = 0, 1, \ldots, d$, $j = 0, 1, \ldots, i$, $k = 0, 1, \ldots, d$, and $l = 0, 1, \ldots, k$,*

$$E^{(d)}_{\frac{i(i+1)}{2}+j+1,\frac{k(k+1)}{2}+l+1} = d^2(d-1)^2 \int\int_T e^{(d)}_{i,j,k,l}(x,y)dxdy,$$

where

$$\begin{aligned} e^{(d)}_{i,j,k,l} = {} & r^2 e^{(d)}_{i,j,k,l} \\ & + rs[e^{(d)}_{i,j,k-1,l} + e^{(d)}_{i-1,j,k,l}] \\ & + rt[e^{(d)}_{i-1,j-1,k,l} + e^{(d)}_{i,j,k-1,l-1}] \\ & + s^2 e^{(d)}_{i-1,j,k-1,l} \\ & + st[e^{(d)}_{i-1,j,k-1,l-1} + e^{(d)}_{i-1,j-1,k-1,l}] \\ & + t^2 e^{(d)}_{i-1,j-1,k-1,l-1} \end{aligned}$$

These recursions begin with the values

$$e^{(2)}_{i,j,k,l} = \frac{1}{2ac}E^{(2)}_{\frac{i(i+1)}{2}+j+1,\frac{k(k+1)}{2}+l+1},$$

contained in the matrix $E^{(2)}$ given above.

Proof: We proceed by induction. Since the area of T is $ac/2$, the assertion is clear for the case $d = 2$. To prove the result for $d > 2$, we use the easily verified recursion formula

$$\Phi_{\alpha,\beta,\gamma}(r,s,t) = r\Phi_{\alpha-1,\beta,\gamma}(r,s,t) + s\Phi_{\alpha,\beta-1,\gamma}(r,s,t) + t\Phi_{\alpha,\beta,\gamma-1}(r,s,t). \qquad (10)$$

Now by (9),

$$p_{xx}(x,y) = d(d-1)\sum_{i=0}^{d}\sum_{j=0}^{i} b_{\frac{i(i+1)}{2}+j+1} A_{ij}^{(d)}(x,y),$$

where

$$A_{i,j}^{(d)}(x,y) = r_x^2\Phi_{d-i-2,i-j,j} + 2r_xs_x\Phi_{d-i-1,i-j-1,j} + 2r_xt_x\Phi_{d-i-1,i-j,j-1}$$
$$+s_x^2\Phi_{d-i,i-j-2,j} + 2s_xt_x\Phi_{d-i,i-j-1,j-1} + t_x^2\Phi_{d-i,i-j,j-2}.$$

Using the recursion relation (10), we find that

$$A_{i,j}^{(d)}(x,y) = rA_{i,j}^{(d-1)}(x,y) + sA_{i-1,j}^{(d-1)}(x,y) + tA_{i-1,j-1}^{(d-1)}(x,y)$$

for $i = 0,1,\ldots,d$ and $j = 0,1,\ldots,i$, provided that we start with $A_{0,0}^{(2)} = r_x^2$, $A_{1,0}^{(2)} = 2r_xs_x$, $A_{1,1}^{(2)} = 2r_xt_x$, $A_{2,0}^{(2)} = s_x^2$, $A_{2,1}^{(2)} = 2s_xt_x$, and $A_{2,2}^{(2)} = t_x^2$. The polynomials p_{xy} and p_{yy} have similar expansions in terms of functions satisfying the same recursions, but with different starting values.

Using the above, we get

$$[p_{xx}]^2 = d^2(d-1)^2\sum_{i=0}^{d}\sum_{j=0}^{i}\sum_{k=0}^{d}\sum_{l=0}^{k} b_{\frac{i(i+1)}{2}+j+1} b_{\frac{k(k+1)}{2}+l+1} e_{xx,i,j,k,l}^{(d)},$$

where

$$e_{xx,i,j,k,l}^{(d)}(x,y) := A_{i,j}^{(d)}(x,y)\cdot A_{k,l}^{(d)}(x,y).$$

Let $e_{xy,i,j,k,l}^{(d)}$ and $e_{yy,i,j,k,l}^{(d)}$ be the relevant expressions involved in the expansions of $[p_{xy}]^2$ and $[p_{yy}]^2$, and let $e_{i,j,k,l}^{(d)} := e_{xx,i,j,k,l}^{(d)} + 2e_{xy,i,j,k,l}^{(d)} + e_{yy,i,j,k,l}^{(d)}$. Then it follows that $e_{i,j,k,l}^{(d)}$ satisfies the stated recursion relation, and

$$[D_{xx}^2 + 2D_{xy}^2 + D_{yy}^2]p = d^2(d-1)^2\sum_{i=0}^{d}\sum_{j=0}^{i}\sum_{k=0}^{d}\sum_{l=0}^{k} b_{\frac{i(i+1)}{2}+j+1} b_{\frac{k(k+1)}{2}+l+1} e_{i,j,k,l}^{(d)}.$$

The result follows since the entries of the energy matrix are obtained by integrating the expressions $e_{i,j,k,l}^{(d)}$ over the triangle T. ■

4. The cubic and quartic cases

In this section we consider cubic and quartic splines as they are the most likely to be useful in practice. In particular, we present the explicit entries in the energy matrices $E^{(3)}$ and $E^{(4)}$. Before proceeding, we observe that by the symmetries involved, in describing $E^{(d)}$, it is not necessary to give all of the $(d+1)^2(d+2)^2/4$ entries. First, by symmetry it suffices to work only with the upper-triangular part of the matrix. Moreover, it is clear that in the upper-triangular part, whenever a formula occurs involving R, S, there are always two others of the same form involving S, T and R, T (cf. the $E_{22}^{(2)}$, $E_{33}^{(2)}$, and $E_{55}^{(2)}$ entries of the matrix $E^{(2)}$ given above). In addition, whenever a formula appears involving either one or three of the letters, then there are always two other similar formulae obtained by cyclic permutation (cf. the $E_{11}^{(2)}$, $E_{44}^{(2)}$, and $E_{66}^{(2)}$, or the $E_{23}^{(2)}$, $E_{25}^{(2)}$, and $E_{35}^{(2)}$ entries of $E^{(2)}$).

The following theorem computes the number of formulae which must be given in order to completely specify the matrix $E^{(d)}$.

Theorem 2. *The number of formulae needed to specify the matrix $E^{(d)}$ is q_d, where $q_{-2} = q_{-1} = 0$, $q_0 = 1$, and where for $k > 0$,*

$$q_{2k} = 2k^3 + 4k + q_{2k-3}$$

$$q_{2k-1} = 2k^3 - 3k^2 + 5k - 2 + q_{2k-4}.$$

Proof: The result follows for $d = 2$ by inspection of the matrix $E^{(2)}$ given above. Now we can proceed by induction. We consider the recursion for q_{2k-1} first. In the first row it is easy to see that it suffices to give $2 + 4 + \cdots + 2k$ entries - this corresponds to the products of b_1 by the b_i associated with domain points in the left half of the triangle T. For example, in the cubic case (cf. Figure 2), we need the $(1,1), (1,2), (1,4), (1,5), (1,7), (1,8)$ entries. Next, for each $i = 1, \ldots, k-1$, in the $\frac{i(i+1)}{2} + 1$-st row of the matrix we need $\frac{(n+1)(n+2)}{2} - 3(2i-1) - 1$ formulae. In the cubic case in Figure 2, these are the $(2,2), (2,3), (2,4), (2.5), (2,6), (2,9)$ entries. This accounts for all rows corresponding to coefficients on the boundary of T. To complete the proof we simply apply the induction hypothesis to the coefficients lying inside the triangle. In the cubic case this is just the $(5,5)$ entry.

The proof of the recursion for q_{2k} is similar. In this case, in the first row it suffices to give $1 + 3 + \cdots + (2k+1)$ entries - this corresponds to the products in the left half of the triangle T. Next, for each $i = 1, \ldots, k-1$, in the $\frac{i(i+1)}{2} + 1$-st row of the matrix we again need $\frac{(n+1)(n+2)}{2} - 3(2i-1) - 1$ formulae. In this case we also need to do the $\frac{k(k+1)}{2} + 1$-st row, which requires $k(k-1) + 2$ formulae. Finally, we apply the induction hypothesis to the coefficients lying inside the triangle. ■

Using Theorem 2 we see that for the cubic case we need 13 formulae, while for the quartic case we need 26 formulae.

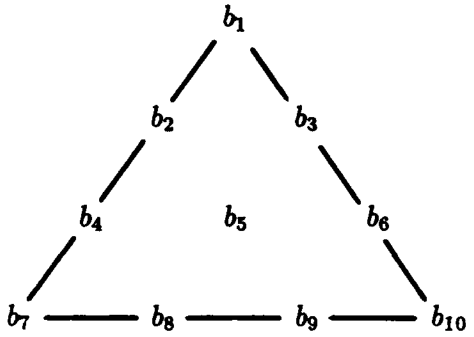


Figure 2. A typical triangle with associated coefficients

Example 3. *The thirteen essential formulae for the entries of* $G = 5E^{(3)}/9ac$ *are*

$$G^{(3)}_{1,1} = (RR)^2$$
$$G^{(3)}_{1,2} = 2RR\,RS + .5(RR)^2$$
$$G^{(3)}_{1,4} = (RS)^2 + RR\,RS$$
$$G^{(3)}_{1,5} = 2RS\,RT + RR\,RT + RR\,RS$$
$$G^{(3)}_{1,7} = .5(RS)^2$$
$$G^{(3)}_{1,8} = RS\,RT + .5(RS)^2$$
$$G^{(3)}_{2,2} = 2RR\,SS + 2(RS)^2 + 2RR\,RS + (RR)^2$$
$$G^{(3)}_{2,3} = 2RR\,ST + 2RS\,RT + RR\,RT + RR\,RS + .5(RR)^2$$
$$G^{(3)}_{2,4} = 2SS\,RS + RR\,SS + 1.5(RS)^2 + 2RR\,RS$$
$$G^{(3)}_{2,5} = 2SS\,RT + 2RS\,ST + RR\,ST + 2RS\,RT + RR\,SS + (RS)^2 + 2RR\,RT + RR\,RS$$
$$G^{(3)}_{2,6} = 2RT\,ST + .5(RT)^2 + RR\,ST + RS\,RT + RR\,RT$$
$$G^{(3)}_{2,9} = RT\,ST + SS\,RT + RS\,ST + (RT)^2 + RS\,RT$$
$$G^{(3)}_{5,5} = 2(SS\,TT + (ST)^2 + TT\,RS + RT\,ST + SS\,RT + RS\,ST + RR\,TT + (RT)^2$$
$$+ RR\,ST + RS\,RT + RR\,SS + (RS)^2)$$

Example 4. *The 26 essential formulae for the entries of $H = 5E^{(4)}/4ac$ are*

$$H^{(4)}_{1,1} = 6(RR)^2$$

$$H^{(4)}_{1,2} = 12RR\,RS + 3(RR)^2$$

$$H^{(4)}_{1,4} = 6(RS)^2 + 6RR\,RS + (RR)^2$$

$$H^{(4)}_{1,5} = 12RS\,RT + 6RR\,RT + 6RR\,RS + (RR)^2$$

$$H^{(4)}_{1,7} = 3(RS)^2 + 2RR\,RS$$

$$H^{(4)}_{1,8} = 6RS\,RT + 3(RS)^2 + 2RR\,RT + 2RR\,RS$$

$$H^{(4)}_{1,11} = (RS)^2$$

$$H^{(4)}_{1,12} = 2RS\,RT + (RS)^2$$

$$H^{(4)}_{1,13} = (RT)^2 + 2RS\,RT + (RS)^2$$

$$H^{(4)}_{2,2} = 12RR\,SS + 12(RS)^2 + 12RR\,RS + 4(RR)^2$$

$$H^{(4)}_{2,3} = 12RR\,ST + 12RS\,RT + 6RR\,RT + 6RR\,RS + 2(RR)^2$$

$$H^{(4)}_{2,4} = 12SS\,RS + 6RR\,SS + 9(RS)^2 + 10RR\,RS + 3(RR)^2$$

$$\begin{aligned}H^{(4)}_{2,5} = {} & 12SS\,RT + 12RS\,ST + 6RR\,ST + 12RS\,RT + 6RR\,SS + 6(RS)^2 + 8RR\,RT + 6RR\,RS \\ & + 2(RR)^2\end{aligned}$$

$$H^{(4)}_{2,6} = 12RT\,ST + 3(RT)^2 + 2RR\,RS + 6RR\,ST + 6RS\,RT + 4RR\,RT + (RR)^2$$

$$H^{(4)}_{2,7} = 6SS\,RS + 6(RS)^2 + 2RR\,SS + 6RR\,RS$$

$$\begin{aligned}H^{(4)}_{2,8} = {} & 6SS\,RT + 6RS\,ST + 6SS\,RS + 10RS\,RT + 2RR\,ST + 2RR\,SS + 4(RS)^2 + 6RR\,RT \\ & + 4RR\,RS\end{aligned}$$

$$\begin{aligned}H^{(4)}_{2,9} = {} & 6RT\,ST + 6SS\,RT + 6RS\,ST + 4(RT)^2 + 2RR\,ST + 6RS\,RT + 2RR\,SS + 2(RS)^2 \\ & + 4RR\,RT + 2RR\,RS\end{aligned}$$

$$H^{(4)}_{2,10} = 6RT\,ST + 2(RT)^2 + 2RR\,ST + 2RS\,RT + 2RR\,RT$$

$$H^{(4)}_{2,13} = 2RT\,ST + 2SS\,RT + 2RS\,ST + 2SS\,RS + 3(RT)^2 + 4RS\,RT + (RS)^2$$

$$H^{(4)}_{2,14} = 2RT\,ST + 2SS\,RT + 2RS\,ST + 2(RT)^2 + 2RS\,RT$$

$$H^{(4)}_{4,4} = 6(SS)^2 + 12SS\,RS + 8RR\,SS + 10(RS)^2 + 12RR\,RS + 6(RR)^2$$

$$\begin{aligned}H^{(4)}_{4,5} = {} & 12SS\,ST + 6SS\,RT + 12RS\,ST + 6SS\,RS + 8RR\,ST + 10RS\,RT + 4RR\,SS + 5(RS)^2 \\ & + 6RR\,RT + 6RR\,RS + 3(RR)^2\end{aligned}$$

$$\begin{aligned}H^{(4)}_{4,6} = {} & 6(ST)^2 + 6RT\,ST + 6RS\,ST + (RT)^2 + 4RR\,ST + 4RS\,RT + (RS)^2 + 2RR\,RT \\ & + 2RR\,RS + (RR)^2\end{aligned}$$

$$\begin{aligned}H^{(4)}_{4,9} = {} & 3(ST)^2 + 6SS\,ST + 8RT\,ST + 4SS\,RT + 6RS\,ST + 2SS\,RS + 3(RT)^2 + 4RR\,ST \\ & + 6RS\,RT + 2RR\,SS + 2(RS)^2 + 6RR\,RT + 2RR\,RS\end{aligned}$$

$$\begin{aligned}H^{(4)}_{5,5} = {} & 12SS\,TT + 12(ST)^2 + 12TT\,RS + 12RT\,ST + 12SS\,RT + 12RS\,ST + 8RR\,TT + 8(RT)^2 \\ & + 8RR\,ST + 12RS\,RT + 8RR\,SS + 8(RS)^2 + 8RR\,RT + 8RR\,RS + 4(RR)^2\end{aligned}$$

$$\begin{aligned}H^{(4)}_{5,6} = {} & 6SS\,TT + 6(ST)^2 + 6SS\,ST + 10TT\,RS + 10RT\,ST + 6SS\,RT + 10RS\,ST + 8SS\,RS \\ & + 6RR\,TT + 6(RT)^2 + 6RR\,ST + 10RS\,RT + 4RR\,SS + 6(RS)^2 + 6RR\,RT + 8RR\,RS\end{aligned}$$

Acknowledgment: The first author was supported in part by the Alexander von Humboldt Foundation, while the second author was supported in part by the National Science Foundation under Grant DMS-8602337.

References

1. Alfeld, P. Triangular extrapolation, University of Wisconsin–Madison Mathematics Research Center Technical Summary Report # 2707, 1984.
2. deBoor, B-form basics, in *Geometric Modeling: Algorithms and New Trends*, SIAM, Philadelphia, 1987, 131–148.
3. Duchon, J., Fonctions–spline homogenes plusieurs variables, Thèse, Grenoble, 1980.
4. Farin, G., Bézier polynomials over triangles and the construction of piecewise C^r polynomials, TR/91, Dept. of Mathematics, Brunel Univ., 1980.
5. Gmelig-Meyling, R., Approximation by cubic C^1 splines on arbitrary triangulations, Numer. Math. **51** (1987), 65–85.
6. von Golitschek, M. and L. L. Schumaker, Data fitting by penalized least squares, this volume.
7. Grandine, T., An iterative method for computing multivariate C^1 piecewise polynomial interpolants, CAGD 4 (1987), 307–319.
8. Schmidt, R., Eine Methode zur Konstruktion von C^1-Flächen zur Interpolation unregelmässig verteilter Daten, in *Multivariate Approximation Theory*, W. Schempp and K. Zeller, eds., Birkhäuser, Basel, 1982, 343–361.
9. Wahba, G., Spline bases, regularization and generalized cross-validation for solving approximation problems with large quantities of noisy data, in *Approximation Theory III*, E. Cheney (ed.), Academic Press, New York, 1980, 905–912.

3. Interpolation

RADIAL BASIS FUNCTION INTERPOLATION ON AN INFINITE REGULAR GRID

M. D. BUHMANN and M. J. D. POWELL
Department of Applied Mathematics and Theoretical Physics
University of Cambridge

Abstract A radial basis function approximation $s(\cdot)$ from $\mathcal{R}^n$ to $\mathcal{R}$ depends on a fixed set of points $\{x_i\}$ in $\mathcal{R}^n$ and on a fixed function $\phi(\cdot)$ from $\mathcal{R}^+$ to $\mathcal{R}$, as it has the form $\{s(x)=\sum_i \lambda_i \phi(\|x-x_i\|_2) \mid x\in\mathcal{R}^n\}$. We consider the case when $\{x_i\}$ is the infinite lattice $\mathcal{Z}^n$ and the coefficients $\{\lambda_i\}$ give interpolation on the lattice. Therefore we study the cardinal function $C(\cdot)$, which is in the space of approximations and satisfies $\{C(\ell)=\delta_{0\ell} \mid \ell\in\mathcal{Z}^n\}$. By employing Fourier transforms, we identify the rate of decay of $|C(x)|$ to zero as $\|x\|\rightarrow\infty$ for several useful choices of $\phi(\cdot)$. Further, an algorithm is presented for calculating $C(x)$ for moderate values of $\|x\|$. It provides some tabulations of the cardinal function that quantify the qualitative asymptotic theory. The results may be highly important to the development of new algorithms for multivariable interpolation, because they include some cases where the asymptotic decay to zero is very fast.

Keywords Cardinal functions. Fourier transforms. Gauss–Seidel iteration. Interpolation. Multivariable approximation. Radial basis functions.

1. Introduction

Radial basis functions provide a versatile family of functions from $\mathcal{R}^n$ to $\mathcal{R}$ that is highly promising for the approximation of functions of several variables in computer calculations. The generic radial basis function, $\phi(\cdot)$ say, is from $\mathcal{R}^+$ to $\mathcal{R}$, and a typical radial basis function approximation with n variables has the form

$$s(x)=\sum_{i=1}^{m}\lambda_i\,\phi(\|x-x_i\|_2),\quad x\in\mathcal{R}^n, \tag{1}$$

where $\{\lambda_i \mid i=1,2,\ldots,m\}$ and $\{x_i \mid i=1,2,\ldots,m\}$ are real parameters and fixed points in $\mathcal{R}^n$ respectively. When interpolating to values of a function $f(\cdot):\mathcal{R}^n\rightarrow\mathcal{R}$, it is usual for the $\{x_i \mid i = 1,2,\ldots,m\}$ to be the points at which the function values are given and for the parameters $\{\lambda_i \mid i = 1,2,\ldots,m\}$ to be defined by the interpolation conditions

$$s(x_i) = f(x_i), \quad i = 1,2,\ldots,m. \tag{2}$$

Of course the points $\{x_i \mid i = 1,2,\ldots,m\}$ should all be different, and then for many useful choices of $\phi(\cdot)$ the interpolation conditions are guaranteed to define the parameters uniquely (Micchelli, 1986). This important work was reviewed at the previous Shrivenham conference (Powell, 1987).

An obvious advantage of expression (1) over the methods that are employed usually for multivariable approximation is that this expression does not become more elaborate as the number of variables is increased. When m is large and $\phi(\cdot)$ does not have finite support, however, then the work of calculating $s(x)$ for any x can be much greater than the work of calculating a typical piecewise polynomial approximation. The purpose of the present paper is to consider another question that is highly important to the usefulness of radial basis function approximation, namely the localization properties of interpolation. In other words, assuming that $s(\cdot)$ can be expressed in the form

$$s(x) = \sum_{i=1}^{m} f(x_i)\, C_i(x), \quad x \in \mathcal{R}^n, \tag{3}$$

where the *cardinal functions* $\{C_i(\cdot) \mid i = 1,2,\ldots,m\}$ are independent of $f(\cdot)$ and satisfy $C_i(x_j) = \delta_{ij}$ in order that the interpolation conditions (2) hold, we ask how quickly $|C_i(x)|$ tends to zero as $\|x - x_i\|$ becomes large. Here $\|\cdot\|$ denotes any norm on $\mathcal{R}^n$. We see that this question is directly relevant to the effect on $s(\cdot)$ of perturbations to the given function values $\{f(x_i) \mid i = 1,2,\ldots,m\}$.

Perhaps the best known example of localization properties without finite support is cubic spline interpolation to values of a function of one variable at equally spaced points throughout the real line. In this case the only zeros of $C_i(\cdot)$ occur at the data points $\{x_j \mid j \neq i\}$, and the cardinal function satisfies the relation

$$C_i(x_i \pm \theta) = (\sqrt{3}-2)\, C_i(x_i \pm [\theta - h]), \quad \theta \geq 2h, \tag{4}$$

where h is the spacing between data points (see Powell, 1981, for instance). Thus, if k is the number of data points between x and x_i, then $|C(x)|$ is bounded above by a multiple of $(2-\sqrt{3})^k$. This example can be construed as radial basis function approximation with $n = 1$ and $\{\phi(r) = r^3 \mid r \in \mathcal{R}^+\}$.

We restrict attention to the case when the set of data points $\{x_i\}$ is the infinite lattice $\mathcal{Z}^n$ of points in $\mathcal{R}^n$ with integer components, in order that several decay properties of cardinal functions can be derived by Fourier techniques. It should be noted, however, that some of the main advantages of radial basis function interpolation occur when the data points are in general position. We let ℓ be the generic point of $\mathcal{Z}^n$, and, using translational symmetry, we write equation (3) in the form

$$s(x) = \sum_{\ell\in\mathcal{Z}^n} f(\ell)\, C(x-\ell), \quad x\in\mathcal{R}^n, \tag{5}$$

where now there is a single cardinal function that satisfies $\{C(j)=\delta_{0j} \mid j\in\mathcal{Z}^n\}$.

Provided that all the sums and integrals in this paragraph are uniformly convergent, it is suitable to let $C(\cdot)$ be the function

$$C(x) = \sum_{\ell\in\mathcal{Z}^n} c_\ell\, \phi(\|x-\ell\|_2), \quad x\in\mathcal{R}^n, \tag{6}$$

where the coefficients have the values

$$c_\ell = \frac{1}{(2\pi)^n}\int_{[-\pi,\pi]^n} \frac{e^{i\ell.\theta}}{\sum_{k\in\mathcal{Z}^n} e^{-ik.\theta}\phi(\|k\|_2)}\, d\theta, \quad \ell\in\mathcal{Z}^n. \tag{7}$$

Here i is $\sqrt{-1}$ and $\ell.\theta$ denotes the scalar product between $\ell\in\mathcal{Z}^n$ and $\theta\in\mathcal{R}^n$. We see that $C(\cdot)$ is in the correct linear space, and that the identity

$$\begin{aligned} C(j) &= \sum_{\ell\in\mathcal{Z}^n} c_\ell\, \phi(\|j-\ell\|_2) \\ &= \sum_{\ell\in\mathcal{Z}^n} c_{j-\ell}\, \phi(\|\ell\|_2) \\ &= \frac{1}{(2\pi)^n}\int_{[-\pi,\pi]^n} \frac{\sum_{\ell\in\mathcal{Z}^n} e^{i(j-\ell).\theta}\phi(\|\ell\|_2)}{\sum_{k\in\mathcal{Z}^n} e^{-ik.\theta}\phi(\|k\|_2)}\, d\theta \\ &= \frac{1}{(2\pi)^n}\int_{[-\pi,\pi]^n} e^{ij.\theta}\, d\theta = \delta_{0j}, \quad j\in\mathcal{Z}^n, \end{aligned} \tag{8}$$

is satisfied, which makes expression (6) plausible. This way of identifying the cardinal function is successful when $\{\phi(r)=e^{-r^2} \mid r\in\mathcal{R}^+\}$, for example, but, due to the conditions of absolute convergence, it cannot be applied directly unless $\sum_{\ell\in\mathcal{Z}^n} |\phi(\|\ell\|_2)|$ is finite. Therefore we employ Fourier methods to study the cardinal functions of the choices

$$\left.\begin{array}{lll} \phi(r) = & r & \text{(linear)} \\ \phi(r) = & r^3 & \text{(cubic)} \\ \phi(r) = & \sqrt{r^2+c^2} & \text{(multiquadric)} \\ \phi(r) = & 1/\sqrt{r^2+c^2} & \text{(inverse multiquadric)} \\ \phi(r) = & r^2\log r & \text{(thin plate spline)} \end{array}\right\}. \tag{9}$$

We are interested in these radial basis functions because each one can give $s\equiv f$ when $f(\cdot)$ is a low order polynomial (Jackson, 1987; Buhmann, 1988b). Specifically, letting Π_q^n be the space of all polynomials from $\mathcal{R}^n$ to $\mathcal{R}$ of total degree at most q, the interpolation formula (5) reproduces each $f(\cdot)\in\Pi_q^n$, where $q=n, n+2, n, n-2$ and $n+1$ in the linear, cubic, multiquadric, inverse multiquadric (assuming $n\geq 2$) and thin plate spline cases respectively. In each of these cases $C(\cdot)$ has the form (6) for certain coefficients $\{c_\ell \mid \ell \in \mathcal{Z}^n\}$ that make the sum (6) absolutely convergent, so the cardinal function is well-defined. Further, the sum (5) is also absolutely convergent for every $x \in \mathcal{R}^n$ and every $f(\cdot) \in \Pi_q^n$, and it gives $\{s(x) = f(x) \mid x \in \mathcal{R}^n\}$ when $f(\cdot) \in \Pi_q^n$. It should be noted, however, that the double sum

$$\sum_{\ell \in \mathcal{Z}^n} f(\ell) \sum_{j \in \mathcal{Z}^n} c_j \, \phi(\|x - \ell - j\|_2), \tag{10}$$

which is obtained by substituting expression (6) in formula (5), need not be absolutely convergent. Indeed, if we were allowed to rearrange terms, then the factor $\sum_{\ell \in \mathcal{Z}^n} f(\ell)\, \phi(\|x-\ell-j\|_2)$ would multiply c_j, but when $f \equiv 1$ this factor is a divergent sum for all of the choices (9).

For example, consider the case when $n=1$ and $\{\phi(r) = r \mid r \in \mathcal{R}^+\}$. The coefficients of the cardinal function (6) have the values $c_0 = -1$, $c_{-1} = c_1 = \frac{1}{2}$ and $\{c_\ell = 0 \mid |\ell| \geq 2\}$ in order that $C(\cdot)$ is the *hat function* of piecewise linear interpolation. Hence formula (5) reproduces all linear polynomials. Thus the space that is spanned by the radial basis functions $\{\, \{\|x - \ell\|_2 \mid x \in \mathcal{R}\} \mid \ell \in \mathcal{Z}\}$ includes Π_1^1, although no nonzero linear polynomial can be written as $\{\sum_{\ell \in \mathcal{Z}} \lambda_\ell \, \|x - \ell\|_2 \mid x \in \mathcal{R}\}$. Therefore, when studying the cases (9) we take the view that the approximation $s(\cdot)$ is to be in the linear space that is spanned by functions of the form (6), which admits the very powerful observation in the previous paragraph that q increases with n. Of course this point of view is unnecessary when m is finite in expression (1). It is also unnecessary when $\{x_i\} \equiv \mathcal{Z}^n$ and $\{\phi(r) = e^{-r^2} \mid r \in \mathcal{R}^+\}$ for example, but in this case the interpolation formula (5) does not give $s \equiv f$ when $f \equiv 1$ (Buhmann, 1988b). Therefore in practice the Gaussian radial basis function is usually far less successful than the nonintegrable functions (9).

We study the cardinal function $C(\cdot)$ theoretically in Section 2. Here we recall from Buhmann (1988b) an explicit expression for the Fourier transform of $C(\cdot)$ that is valid for all of the choices (9), and we also recall that the asymptotic decay rate of $|C(x)|$ to zero as $\|x\| \to \infty$ can be deduced from the differentiability properties of this transform. This analysis identifies the dominant term of $C(x)$ for large $\|x\|$ when $\phi(r) = r$ and $n = 2$.

In Section 3 we present a procedure for calculating values of $C(x)$. It employs a quasi-interpolating function

$$\psi(x) = \sum_{\ell \in \hat{\mathcal{Z}}} \gamma_\ell \, \phi(\|x - \ell\|_2), \quad x \in \mathcal{R}^n, \tag{11}$$

where $\hat{\mathcal{Z}}$ is a finite subset of $\mathcal{Z}^n$, that ideally possesses the diagonal dominance condition

$$|\psi(0)| > {\sum_{\ell \in \mathcal{Z}^n}}' |\psi(\ell)|. \tag{12}$$

Here and throughout this paper the prime on the summation indicates that the $\ell=0$ term is omitted. This function is used in an iteration of Gauss–Seidel type to obtain an approximation $C_b(\cdot)$ to $C(\cdot)$ that satisfies the cardinality equations

$$C_b(\ell) = \delta_{0\ell}, \quad \ell \in \mathcal{Z}^n, \quad \|\ell\|_\infty \leq b, \tag{13}$$

for some fixed integer b. We find that this approximation is sufficiently accurate to give good estimates of the true cardinal function for $\|x\|_\infty \leq \frac{1}{2}b$.

Thus several tables and graphs were calculated that show properties of $C(x)$ for $n=2$ and $n=3$. We find when $n=3$ and $\phi(r)=r$, for example, that $\{|C(x)| < 10^{-6} \mid \|x\|_\infty > 5\}$, which is superior to the decay rate (4) of cubic spline interpolation. These numerical results are given and discussed in Section 4, and we note that they suggest several topics for further research.

2. The Fourier transform and decay properties of the cardinal function

The Fourier transform of a continuous and absolutely integrable function $\{g(x) \mid x \in \mathcal{R}^n\}$ is defined by the formula

$$\hat{g}(t) = \int_{\mathcal{R}^n} e^{-ix.t}\, g(x)\, dx, \quad t \in \mathcal{R}^n, \tag{14}$$

and, when $\hat{g}(\cdot)$ is continuous and absolutely integrable too, the inverse formula is the relation

$$g(x) = \frac{1}{(2\pi)^n} \int_{\mathcal{R}^n} e^{ix.t}\, \hat{g}(t)\, dt, \quad x \in \mathcal{R}^n, \tag{15}$$

(see Stein and Weiss, 1971, for instance). In this section we derive some decay properties of the cardinal function $C(\cdot)$ from its Fourier transform $\hat{C}(\cdot)$ for each of the radial basis functions (9).

It is proved by Buhmann (1988b) that in all these cases $\hat{C}(\cdot)$ has the remarkably simple form

$$\hat{C}(t) = \frac{\hat{\phi}(\|t\|_2)}{\sum_{k \in \mathcal{Z}^n} \hat{\phi}(\|t - 2\pi k\|_2)}, \quad t \in \mathcal{R}^n, \tag{16}$$

where $\{\hat{\phi}(\|t\|_2) \mid t \in \mathcal{R}^n\}$ is the *generalized Fourier transform* of the radially symmetric function $\{\phi(\|x\|_2) \mid x \in \mathcal{R}^n\}$. Jones (1982) presents an excellent introduction to generalized transforms, and shows that the functions

$$\left.\begin{array}{lll}
\hat{\phi}(r) = & r^{-n-1} & \text{(linear)} \\
\hat{\phi}(r) = & r^{-n-3} & \text{(cubic)} \\
\hat{\phi}(r) = & (c/r)^{(n+1)/2} K_{(n+1)/2}(cr) & \text{(multiquadric)} \\
\hat{\phi}(r) = & (c/r)^{(n-1)/2} K_{(n-1)/2}(cr) & \text{(inverse multiquadric)} \\
\hat{\phi}(r) = & r^{-n-2} & \text{(thin plate spline)}
\end{array}\right\} \tag{17}$$

are appropriate to the radial basis functions (9), except that for simplicity we have suppressed some constant normalization factors because they are irrelevant to the ratio (16). Here $K_\nu(\cdot)$ is the ν-th modified Bessel function as defined in Abramowitz and Stegun (1970).

We see that in all these cases the generalized Fourier transform $\{\hat{\phi}(\|t\|_2) \mid t \in \mathcal{R}^n\}$ is unbounded at the origin, but otherwise it is positive, continuous and absolutely integrable over every closed part of $\mathcal{R}^n$ that excludes the origin. In view of equations (16) and (17), this unboundedness implies the values $\hat{C}(0)=1$ and $\{\hat{C}(2\pi k)=0 \mid k \in \mathcal{Z}^n\backslash\{0\}\}$, which are important to the fact that interpolation reproduces low order polynomials (Jackson, 1987). Further, when t is not in the set of lattice points $\{2\pi k \mid k \in \mathcal{Z}^n\}$, the denominator of expression (16) is an absolutely convergent sum that is periodic and bounded away from zero. It follows from all these remarks that $C(\cdot)$ is well defined when we substitute $\hat{g}(\cdot)=\hat{C}(\cdot)$ in the inverse Fourier transform formula (15).

Buhmann (1988b) uses some properties of generalized functions to prove that this $C(\cdot)$ is the required cardinal function, but the following simple analysis is sufficient when certain sums and integrals are absolutely convergent, including the conditions that $\{\phi(\|x\|_2) \mid x \in \mathcal{R}^n\}$ is the classical inverse Fourier transform of $\{\hat{\phi}(\|t\|_2) \mid t \in \mathcal{R}^n\}$ and $\sum_{k\in\mathcal{Z}^n} \hat{\phi}(\|t-2\pi k\|_2)$ is nonzero for all t. For example, these conditions hold for the Gaussian radial basis function $\{\phi(r)=e^{-r^2} \mid r \in \mathcal{R}^+\}$ because $\{\hat{\phi}(\|t\|_2) \mid t \in \mathcal{R}^n\}$ is a constant multiple of $\{e^{-\|t\|_2^2/4} \mid t \in \mathcal{R}^n\}$. Corresponding to the identity (8), equations (15) and (16) give the cardinality conditions

$$\begin{aligned} C(j) &= \frac{1}{(2\pi)^n}\int_{\mathcal{R}^n} \frac{e^{ij.t}\,\hat{\phi}(\|t\|_2)}{\sum_{k\in\mathcal{Z}^n}\hat{\phi}(\|t-2\pi k\|_2)}\,dt \\ &= \frac{1}{(2\pi)^n}\int_{[-\pi,\pi]^n} \frac{\sum_{\ell\in\mathcal{Z}^n} e^{ij.(t-2\pi\ell)}\,\hat{\phi}(\|t-2\pi\ell\|_2)}{\sum_{k\in\mathcal{Z}^n}\hat{\phi}(\|t-2\pi k\|_2)}\,dt \\ &= \frac{1}{(2\pi)^n}\int_{[-\pi,\pi]^n} e^{ij.t}\,dt \;=\; \delta_{0j}, \quad j\in\mathcal{Z}^n, \end{aligned} \tag{18}$$

where the middle line depends on the periodicity of the denominator and the last line on $e^{ij.(2\pi\ell)}=1$. Moreover, to show that $C(\cdot)$ has the form (6), we note that the continuous periodic function

$$\omega(t) = 1 \Big/ \sum_{k\in\mathcal{Z}^n} \hat{\phi}(\|t-2\pi k\|_2), \quad t\in\mathcal{R}^n, \tag{19}$$

is equal to its Fourier expansion, so we have the identity

$$\omega(t) = \sum_{\ell\in\mathcal{Z}^n} c_\ell\, e^{-i\ell.t}, \tag{20}$$

where the coefficients have the values

$$c_\ell = \frac{1}{(2\pi)^n}\int_{[-\pi,\pi]^n} e^{i\ell.t}\,\omega(t)\,dt, \quad \ell\in\mathcal{Z}^n. \tag{21}$$

Thus $C(\cdot)$ is the function

$$\begin{aligned} C(x) &= \frac{1}{(2\pi)^n}\int_{\mathcal{R}^n} e^{ix.t}\,\hat{C}(t)\,dt \\ &= \frac{1}{(2\pi)^n}\int_{\mathcal{R}^n} e^{ix.t}\,\omega(t)\,\hat{\phi}(\|t\|_2)\,dt \\ &= \sum_{\ell\in\mathcal{Z}^n} c_\ell\,\{\frac{1}{(2\pi)^n}\int_{\mathcal{R}^n} e^{i(x-\ell).t}\,\hat{\phi}(\|t\|_2)\,dt\} \\ &= \sum_{\ell\in\mathcal{Z}^n} c_\ell\,\phi(\|x-\ell\|_2), \quad x\in\mathcal{R}^n, \end{aligned} \tag{22}$$

where the third and fourth lines depend on equations (20) and (15) respectively. This expression is identical to equation (6) as required. Further, it can be deduced from the ***Poisson summation formula*** (see Stein and Weiss, 1971, for instance) that the coefficients (7) and (21) are the same.

Next we consider the localization properties of interpolation that can be deduced from $\hat{C}(\cdot)$. If $\{\hat{C}(t) \mid t\in\mathcal{R}^n\}$ has absolutely integrable second derivatives and is sufficiently smooth for large $\|t\|$, then integration by parts applied to formula (15) gives the relation

$$\begin{aligned} C(x) &= \frac{1}{(2\pi)^n}\int_{\mathcal{R}^n} e^{ix.t}\,\hat{C}(t)\,dt \\ &= \frac{-1}{(2\pi)^n\,(ix_1)}\int_{\mathcal{R}^n} e^{ix.t}\,[\frac{d}{dt_1}\hat{C}(t)]\,dt \\ &= \frac{-1}{(2\pi)^n\,x_1^2}\int_{\mathcal{R}^n} e^{ix.t}\,[\frac{d^2}{dt_1^2}\hat{C}(t)]\,dt, \end{aligned} \tag{23}$$

which implies the identity

$$\|x\|_2^2\,C(x) = \frac{-1}{(2\pi)^n}\int_{\mathcal{R}^n} e^{ix.t}\,[\nabla^2\hat{C}(t)]\,dt, \quad x\in\mathcal{R}^n, \tag{24}$$

and there is an upper bound on the modulus of the right hand side that is independent of x. Similarly, if $\hat{C}(\cdot)$ has absolutely integrable derivatives of order j and enough smoothness at infinity, this technique provides the inequality

$$|C(x)| \le \Lambda_j\,[1+\|x\|_2]^{-j}, \quad x\in\mathcal{R}^n, \tag{25}$$

for some positive constant Λ_j. Thus the decay properties of the cardinal function $C(\cdot)$ are intimately related to the differentiability of its Fourier transform $\hat{C}(\cdot)$.

Therefore, as in Buhmann (1988b), we consider the differentiability of the function (16) for each of the choices (17) of $\hat{\phi}(\cdot)$. In the linear, thin plate spline and cubic cases we have the Fourier transform

$$\hat{C}(t) = [\,1+\|t\|_2^{n+m}\sum_{k\in\mathcal{Z}^n}{}'\|t-2\pi k\|_2^{-n-m}\,]^{-1}, \quad t\in\mathcal{R}^n, \tag{26}$$

where $m=1,2$ and 3 respectively. We see that this function is infinitely differentiable throughout $\mathcal{R}^n$ when $n+m$ is even, which gives the important conclusion that a bound of the form (25) is satisfied for every positive j. In fact in these cases $|C(x)|$ decays exponentially as $\|x\| \to \infty$ (Madych and Nelson, 1987; Buhmann, 1988b), which means that the condition

$$|C(x)| \leq \Lambda\, \mu^{\|x\|}, \quad x \in \mathcal{R}^n, \tag{27}$$

is satisfied for some constants $\Lambda > 0$ and $\mu < 1$. One purpose of the numerical results of Section 4 is to indicate the magnitude of μ when $n=2$ and $n=3$.

When $n+m$ is odd in expression (26), however, then $\hat{C}(\cdot)$ has singularities at the grid points $t \in 2\pi\mathcal{Z}^n$ that restrict the value of j in inequality (25). The singularities at $t=0$ and when $t \in \{2\pi k \mid k \in \mathcal{Z}^n \backslash \{0\}\}$ are shown in the expansions

$$\hat{C}(t) = 1 - \|t\|_2^{n+m} \sum_{k\in\mathcal{Z}^n}{}' \|2\pi k\|_2^{-n-m} + O(\|t\|_2^{n+m+1}) \tag{28}$$

and

$$\hat{C}(t) = \|t-2\pi k\|_2^{n+m}\, \|2\pi k\|_2^{-n-m} + O(\|t-2\pi k\|_2^{n+m+1}) \tag{29}$$

respectively, and it is the contribution from these singularities to $\{C(x) \mid x \in \mathcal{R}^n\}$ through the first line of equation (23) that dominates the magnitude of $|C(x)|$ for large $\|x\|$. Instead of integrating by parts, we pick out this contribution from the fact that the generalized inverse transform of $\{\|t\|_2^{n+m} \mid t \in \mathcal{R}^n\}$ when $n+m$ is odd is the function

$$\frac{2^{n+m}\,\Gamma(n+\frac{1}{2}m)}{\pi^{n/2}\,\Gamma(-\frac{1}{2}n-\frac{1}{2}m)}\, \|x\|_2^{-2n-m}, \quad x \in \mathcal{R}^n. \tag{30}$$

Specifically, the terms from the singularities of expressions (28) and (29) are given explicitly in the equation

$$\begin{aligned} C(x) &= \frac{1}{(2\pi)^n} \int_{\mathcal{R}^n} e^{ix.t}\, \hat{C}(t)\, dt \\ &= \frac{1}{(2\pi)^n} \int_{\mathcal{R}^n} e^{ix.t} [\, -\|t\|_2^{n+m} \sum_{k\in\mathcal{Z}^n}{}' \|2\pi k\|_2^{-n-m} + \cdots \\ &\quad + \sum_{k\in\mathcal{Z}^n}{}' \|t-2\pi k\|_2^{n+m}\, \|2\pi k\|_2^{-n-m} + \cdots]\, dt \\ &= \frac{1}{(2\pi)^n} \int_{\mathcal{R}^n} e^{ix.t} [\, -\|t\|_2^{n+m} \sum_{k\in\mathcal{Z}^n}{}' \|2\pi k\|_2^{-n-m} + \cdots] \\ &\quad + [\, \sum_{k\in\mathcal{Z}^n}{}' e^{ix.(t+2\pi k)}\, \|t\|_2^{n+m}\, \|2\pi k\|_2^{-n-m} + \cdots]\, dt \\ &= \frac{1}{(2\pi)^n} \int_{\mathcal{R}^n} e^{ix.t}\, \|t\|_2^{n+m} \sum_{k\in\mathcal{Z}^n}{}' [-1 + e^{2\pi i x.k}]\, \|2\pi k\|_2^{-n-m} \; + \cdots\, dt \\ &= \frac{2^{n+m}\,\Gamma(n+\frac{1}{2}m)}{\pi^{n/2}\,\Gamma(-\frac{1}{2}n-\frac{1}{2}m)} \{ \sum_{k\in\mathcal{Z}^n}{}' [-1 + e^{2\pi i x.k}]\, \|2\pi k\|_2^{-n-m} \}\, \|x\|_2^{-2n-m} \\ &\quad + o(\|x\|_2^{-2n-m}), \end{aligned} \tag{31}$$

which is valid for all large $\|x\|$. We see that the sum in the last line of this expression is absolutely convergent, real, nonpositive and periodic in $x \in \mathcal{R}^n$. Further, it vanishes at the grid points $x \in \mathcal{Z}^n$ in accordance with the cardinality conditions $\{C(\ell) = \delta_{0\ell} \mid \ell \in \mathcal{Z}^n\}$.

Later we give particular attention to the linear radial basis function $\phi(r) = r$ when $n=2$ and $m=1$. Therefore we note that in this case expression (30) has the value $9/(2\pi\|x\|_2^5)$, and for any x it is easy to work out numerically the sum in the last line of equation (31). Thus we find the relations

$$\left.\begin{array}{ll} C(x) \approx -0.05747\,\|x\|_2^{-5}, & [x + (\tfrac{1}{2}, 0)] \in \mathcal{Z}^2 \\ C(x) \approx -0.06734\,\|x\|_2^{-5}, & [x + (\tfrac{1}{2}, \tfrac{1}{2})] \in \mathcal{Z}^2 \end{array}\right\} \tag{32}$$

for the values of the cardinal function at the midpoints of edges of grid squares and at the midpoints of grid squares respectively. These asymptotic results agree well with the calculations of Section 4 for moderate values of $\|x\|$.

The following argument shows that, for even n, the asymptotic decay properties of $C(\cdot)$ in the multiquadric and inverse multiquadric cases are similar to expression (31). The dominant singularity of $\{K_\nu(\theta) \mid \theta \in \mathcal{R}^+\}$ at the origin is a multiple of $\theta^{-\nu}$, so we let $m=1$ and $m=-1$ for multiquadrics and inverse multiquadrics respectively, in order that lines 3 and 4 of expression (17) give the relation

$$\hat{\phi}(r) = \mu_m\, r^{-n-m} + O(r^{-n-m+1}), \quad r \in \mathcal{R}^+, \tag{33}$$

where μ_m is a constant. It follows from formula (16) that, instead of equations (28) and (29), we now have the expansions

$$\hat{C}(t) = 1 - \mu_m^{-1}\, \|t\|_2^{n+m} \sum_{k \in \mathcal{Z}^n}{}' \hat{\phi}(\|2\pi k\|_2) + O(\|t\|_2^{n+m+1}) \tag{34}$$

and

$$\hat{C}(t) = \mu_m^{-1}\, \|t - 2\pi k\|_2^{n+m}\, \hat{\phi}(\|2\pi k\|_2) + O(\|t - 2\pi k\|_2^{n+m+1}). \tag{35}$$

Thus expression (31) is still valid except that the sum inside the braces is replaced by the term

$$\mu_m^{-1} \sum_{k \in \mathcal{Z}^n}{}' [-1 + e^{2\pi i x.k}]\, \hat{\phi}(\|2\pi k\|_2). \tag{36}$$

In order to be more explicit, we recall from Abramowitz and Stegun (1970) that, when $\nu + \frac{1}{2}$ is a positive integer, the ν-th modified Bessel function is the finite sum

$$K_\nu(\theta) = \sqrt{\pi}\, e^{-\theta} \sum_{q=0}^{\nu-\frac{1}{2}} \frac{(2\nu - q - 1)!}{q!\,(\nu - q - \frac{1}{2})!}\, (2\theta)^{q-\nu}, \quad \theta \in \mathcal{R}^+, \tag{37}$$

which after some manipulation gives the value

$$\frac{\hat{\phi}(\|2\pi k\|_2)}{\mu_m} = \|2\pi k\|_2^{-n-m}\, e^{-\|2\pi c k\|_2} \sum_{q=0}^{\frac{1}{2}(n+m-1)} \frac{(\frac{1}{2}[n+m-1])!\,(n+m-q-1)!\,\|4\pi c k\|_2^q}{(\frac{1}{2}[n+m-1]-q)!\,(n+m-1)!\,q!}. \tag{38}$$

When this expression replaces $\|2\pi k\|_2^{-n-m}$ in equation (31), we have the dominant asymptotic term of the cardinal function $C(\cdot)$ for the multiquadric ($m=1$) and inverse multiquadric ($m=-1$) radial basis functions, provided that n is even.

In particular, taking $n=2$, $m=1$, $c=\frac{1}{2}$ and x at the midpoint of an edge of a grid square, this analysis gives the relation

$$\begin{aligned} C(x) &= \frac{9}{2\pi}\sum_{k\in\mathcal{Z}^2}{}'[-1+e^{2\pi i x.k}]\,[\frac{1}{\|2\pi k\|_2^3}+\frac{c}{\|2\pi k\|_2^2}]\,e^{-c\|2\pi k\|_2}\|x\|_2^{-5}+o(\|x\|_2^{-5}) \\ &= -0.005212\,\|x\|_2^{-5}+o(\|x\|_2^{-5}), \quad [x+(\tfrac{1}{2},0)]\in\mathcal{Z}^2, \end{aligned} \tag{39}$$

when $\|x\|$ is large. One unexpected conclusion from equations (32) and (39) is that in two dimensions multiquadric interpolation can have better localization properties than linear interpolation ($\phi(r)=r$).

If n is odd, however, then neither multiquadrics nor inverse multiquadrics give the exponential decay (27). In these cases $n+m$ is even in equation (33), so the r^{-n-m} term does not cause a singularity in the Fourier transform (16) of the cardinal function, but there is some loss of differentiability due to the expansion of the modified Bessel function $\{K_{(n+m)/2}(cr) \mid r\in\mathcal{R}^+\}$ at $r=0$. Specifically, the dominant term of $\hat{\phi}(r)$ for small r and the first term that gives a singularity are $\mu_m r^{-n-m}$ and $\hat{\mu}_m \log r$ respectively, where μ_m and $\hat{\mu}_m$ are constants. Now equation (16) implies the relation

$$\begin{aligned} \hat{C}(t) &\approx 1-[\hat{\phi}(\|t\|_2)]^{-1}\sum_{k\in\mathcal{Z}^n}{}'\hat{\phi}(\|2\pi k\|_2) \\ &\approx 1-\mu_m^{-1}\,\|t\|_2^{n+m}[1+\cdots+\frac{\hat{\mu}_m}{\mu_m}\,\|t\|_2^{n+m}\,\log\|t\|_2+\cdots]^{-1}\sum_{k\in\mathcal{Z}^n}{}'\hat{\phi}(\|2\pi k\|_2) \end{aligned} \tag{40}$$

for small t. Thus the leading singularity in $\hat{C}(\cdot)$ is proportional to $\{\|t\|_2^{2n+2m}\log\|t\|_2 \mid t\in\mathcal{R}^n\}$. The generalized inverse Fourier transform of this expression is a multiple of $\{\|x\|_2^{-3n-2m} \mid x\in\mathcal{R}^n\}$. Hence, for odd n and large $\|x\|$, $|C(x)|$ is $O(\|x\|_2^{-3n-2})$ and $O(\|x\|_2^{-3n+2})$ for the multiquadric and inverse multiquadric radial basis functions respectively.

3. The calculation of cardinal functions

The results of Jackson (1988) suggest that, when $n=3$ and $\{\phi(r)=r \mid r\in\mathcal{R}^+\}$, it is possible for a quasi-interpolating function

$$\psi(x)=\sum_{\ell\in\hat{\mathcal{Z}}}\gamma_\ell\,\phi(\|x-\ell\|_2), \quad x\in\mathcal{R}^n, \tag{41}$$

where $\hat{\mathcal{Z}}$ is a finite subset of $\mathcal{Z}^n$ and $\{\gamma_\ell \mid \ell \in \hat{\mathcal{Z}}\}$ is a set of constant coefficients, to possess the diagonal dominance condition

$$|\psi(0)| > \sum_{\ell\in\mathcal{Z}^n}{}' |\psi(\ell)|. \tag{42}$$

In this case, with the normalization $\psi(0)=1$, one can deduce from the Gauss–Seidel iteration

$$C^{[k+1]}(x) = C^{[k]}(x) - \{C^{[k]}(j) - \delta_{0j}\}\, \psi(x-j), \quad x \in \mathcal{R}^n, \tag{43}$$

j being a vector from $\mathcal{Z}^n$ that depends on k, that a cardinal function of interpolation exists. One uses this formula to generate a convergent sequence of approximating functions $\{C^{[k+1]}(\cdot) \mid k = 1,2,3,\ldots\}$ whose limit satisfies $\{C^{[\infty]}(\ell) = \delta_{0\ell} \mid \ell \in \mathcal{Z}^n\}$. This technique is mentioned because in this section we present a procedure for the calculation of approximations to cardinal functions that includes an iteration of Gauss–Seidel type that is derived from equation (43).

Our procedure begins by generating a quasi-interpolating function of the form (41) in a way that is described later. In order to take advantage of symmetry, we let $\hat{\mathcal{Z}} = \mathcal{Z}_q^n$ for some small integer q (usually $q=3$ or $q=4$), where we define $\mathcal{Z}_q^n$ to be the set $\mathcal{Z}^n \cap [-q,q]^n$. Thus the coefficients of expression (41) are chosen to satisfy $\gamma_{P\ell} = \gamma_\ell$, where $P\ell$ is any vector in $\mathcal{Z}^n$ whose components have moduli that are a permutation of the moduli of the components of ℓ. In this case we say that $P\ell$ and ℓ are *equivalent*. Each function $C^{[k]}(\cdot)$ has the form

$$C^{[k]}(x) = \sum_{\ell\in\mathcal{Z}_b^n} \mu_\ell^{[k]}\, \psi(x-\ell), \quad x \in \mathcal{R}^n, \tag{44}$$

a typical value of the integer b being $b=20$, and each integer vector that is analogous to j in equation (43) is restricted to the set $\mathcal{Z}_b^n$. Therefore in the limit $k \to \infty$ we try to achieve the conditions

$$C^{[\infty]}(\ell) = \delta_{0\ell}, \quad \ell \in \mathcal{Z}_b^n, \tag{45}$$

which correspond to the equations (13) with $C_b(\cdot) \equiv C^{[\infty]}(\cdot)$. We see that expressions (44) and (45) give a $(2b+1)^n \times (2b+1)^n$ system of linear equations in the coefficients $\{\mu_\ell^{[\infty]} \mid \ell \in \mathcal{Z}_b^n\}$ that is to be solved by a Gauss–Seidel iteration. We take advantage of the equivalence symmetry $\mu_{P\ell}^{[\infty]} = \mu_\ell^{[\infty]}$ to reduce the number of unknowns to $\frac{1}{2}(b+1)(b+2)$ and $\frac{1}{6}(b+1)(b+2)(b+3)$ when $n=2$ and $n=3$ respectively, which are the only values of n in the numerical results of Section 4.

To begin the iteration we set $C^{[1]}(\cdot) = \psi(\cdot)$ and $k=1$. For each k the vector $j \in \mathcal{Z}_b^n$ is chosen to maximize the residual $\{|C^{[k]}(j) - \delta_{0j}| \mid j \in \mathcal{Z}_b^n\}$. If this maximum residual is less than a preset tolerance the calculation ends, and from now on we use the notation $C_b(\cdot)$ to denote the final $C^{[k]}(\cdot)$. Otherwise a step of Gauss–Seidel type is applied to give $C^{[k+1]}(\cdot)$, but equation (43) is not used as it stands because it would lose the equivalence symmetry. Instead we define $\mathcal{P}(j)$ to be the subset of $\mathcal{Z}_b^n$ that is composed of j and all other vectors that are equivalent to j, and we let $C^{[k+1]}(\cdot)$ be the function

$$C^{[k+1]}(x) = C^{[k]}(x) - \{C^{[k]}(j) - \delta_{0j}\} \sum_{m\in\mathcal{P}(j)} \psi(x-m), \quad x\in\mathcal{R}^n. \qquad (46)$$

Because the symmetry implies $\{C^{[k]}(m) = C^{[k]}(j) \mid m\in\mathcal{P}(j)\}$ and because $\psi(0)=1$, it can be shown that this iteration gives the inequality

$$\begin{aligned} \sum_{\ell\in\mathcal{Z}_b^n} |C^{[k+1]}(\ell) - \delta_{0\ell}| \;\le\; & \sum_{\ell\in\mathcal{Z}_b^n\backslash\mathcal{P}(j)} |C^{[k]}(\ell) - \delta_{0\ell}| \\ & + \sum_{m\in\mathcal{P}(j)} |C^{[k]}(m) - \delta_{0m}| \sum_{\ell\in\mathcal{Z}_b^n\backslash\{m\}} |\psi(\ell-m)|. \qquad (47) \end{aligned}$$

Therefore convergence is guaranteed if condition (42) holds. The iterative procedure terminates quite efficiently in all of the calculations that are mentioned in the next section, but the diagonal dominance (42) is not obtained in some of these cases.

Several of these calculations require more than 24 hours of running time on a Sun 3/50 workstation, because of the number of terms that occur. For example, when $n=3$, $b=20$ and $\hat{\mathcal{Z}} = \mathcal{Z}_3^3$, the computation of $C_b(x)$ for general x requires 103,823 different values of $\phi(r)$ to be determined and each one involves a square root or a logarithm. It is therefore important to plan the details of the algorithm carefully. The following features are included in our iterative procedure.

Before beginning the iterations we calculate the values $\{\psi(\ell) \mid \ell\in\mathcal{Z}_{2b}^n\}$ of the quasi-interpolating function (41), and for each k we have available the coefficients $\{\mu_\ell^{[k]} \mid \ell\in\mathcal{Z}_b^n\}$, the residuals $\{C^{[k]}(\ell) - \delta_{0\ell} \mid \ell\in\mathcal{Z}_b^n\}$ and the value of ℓ that gives the residual of maximum modulus, which is the j of formula (46). In view of this formula, the new coefficients $\{\mu_\ell^{[k+1]} \mid \ell\in\mathcal{Z}_b^n\backslash\mathcal{P}(j)\}$ are the same as the old ones $\{\mu_\ell^{[k]} \mid \ell\in\mathcal{Z}_b^n\backslash\mathcal{P}(j)\}$ but the $\{\mu_\ell^{[k]} \mid \ell\in\mathcal{P}(j)\}$ are overwritten by $\{\mu_\ell^{[k]} - \rho^{[k]} \mid \ell\in\mathcal{P}(j)\}$ where $\rho^{[k]} = C^{[k]}(j) - \delta_{0j}$. Further, the residuals are overwritten by the numbers

$$\{C^{[k+1]}(\ell) - \delta_{0\ell}\} = \{C^{[k]}(\ell) - \delta_{0\ell}\} - \rho^{[k]} \sum_{m\in\mathcal{P}(j)} \psi(\ell-m), \quad \ell\in\mathcal{Z}_b^n, \qquad (48)$$

and during this process the next value of j is determined. Because the values of $\psi(\cdot)$ in equation (48) are available explicitly, there are no calculations of radial basis functions during the iterations and the amount of work per iteration is independent of the number of terms in $\hat{\mathcal{Z}} = \mathcal{Z}_q^n$. In all of these operations we take advantage of the equivalence symmetry. The computer program stores and updates only one of the coefficients $\{\mu_\ell^{[k]}\}$ and one of the residuals $\{C^{[k]}(\ell) - \delta_{0\ell}\}$ for each of the equivalence sets $\mathcal{P}(\ell)$, but, in order to save some work when equation (48) is applied, it stores $\psi(\ell)$ for every ℓ in $\mathcal{Z}^n \cap [0,2b]^n$.

Another feature of this program is that it allows a range of values of b. They are treated in ascending order, and we set $C^{[1]}(\cdot) \equiv \psi(\cdot)$ only when b is least. For each b the iterations continue until the termination condition gives $C_b(\cdot)$, and this approximation to the cardinal function is chosen as the $C^{[1]}(\cdot)$ of the next b. This feature is useful because comparing tabulations of $C_b(\cdot)$ for several values of b provides a good indication of the accuracy of these approximations to the cardinal function. An example of such a comparison is considered later.

The coefficients $\{\gamma_\ell \mid \ell \in \mathcal{Z}_q^n\}$ of the quasi-interpolating function (41) are determined by a linear programming procedure that tries to achieve inequality (42) subject to the normalization condition

$$\psi(0) = 1. \tag{49}$$

Because the sum (42) is infinite, a finite integer a satisfying $a \geq q$ has to be chosen. Then the coefficients are calculated to minimize the objective function

$$\sum_{j \in \mathcal{Z}_a^n} |\psi(j)| = \sum_{j \in \mathcal{Z}_a^n} | \sum_{\ell \in \mathcal{Z}_q^n} \gamma_\ell \, \phi(\|j - \ell\|_2) |. \tag{50}$$

Except in the $n=2$ inverse multiquadric case, condition (49) is augmented by further linear equality constraints that help $|\psi(x)|$ to become small as $\|x\| \rightarrow \infty$.

Most of these constraints are contained in the statement that, except in the $n=2$ inverse multiquadric case, $\psi(\cdot)$ shall have the form

$$\psi(x) = \sum_{\ell \in \mathcal{Z}_{q-1}^n} \hat{\gamma}_\ell \{\sum_{i=1}^n \delta_i^2 \phi(\|x - \ell\|_2)\}, \quad x \in \mathcal{R}^n, \tag{51}$$

where δ_i^2 is the central difference operator

$$\delta_i^2 \phi(\|x - \ell\|_2) = \phi(\|x - e_i - \ell\|_2) - 2\phi(\|x - \ell\|_2) + \phi(\|x + e_i - \ell\|_2), \tag{52}$$

e_i being the unit vector along the i-th coordinate direction of $\mathcal{R}^n$. In addition to helping the boundedness of $\psi(\cdot)$, expression (51) has the strong advantage in the linear, cubic and multiquadric cases that some cancellation can be done analytically when $\psi(x)$ is computed for large $\|x\|$. For example, the identity

$$\begin{aligned}
&\{(x_1-1)^2+\alpha^2\}^{\frac{1}{2}} - 2\{x_1^2+\alpha^2\}^{\frac{1}{2}} + \{(x_1+1)^2+\alpha^2\}^{\frac{1}{2}} \\
&= 8\,\alpha^2 \,/\, (\,[\,\{(x_1-1)^2+\alpha^2\}^{\frac{1}{2}}\,\{(x_1+1)^2+\alpha^2\}^{\frac{1}{2}} + x_1^2 + \alpha^2 - 1\,] \\
&\qquad \times [\,\{(x_1-1)^2+\alpha^2\}^{\frac{1}{2}} + 2\,\{x_1^2+\alpha^2\}^{\frac{1}{2}} + \{(x_1+1)^2+\alpha^2\}^{\frac{1}{2}}\,]\,)
\end{aligned} \tag{53}$$

is highly useful for large x_1 when $\phi(r) = r$ or $\sqrt{(r^2+c^2)}$. Such identities improve the accuracy of many of the numerical results of Section 4, particularly when $\phi(r) = r$ and $n=3$, because in this case some values of the cardinal function are given whose moduli are less than 10^{-10}.

We also include the constraints

$$\sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell = \sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell \, (\ell_1^4 + \ell_2^4 + \ell_1^2 + \ell_2^2 - 6\, \ell_1^2 \ell_2^2) = 0, \tag{54}$$

$$\sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell = \sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell \, (\ell_1^4 + \ell_2^4 + \ell_3^4 + \ell_1^2 + \ell_2^2 + \ell_3^2 - 3\, [\ell_1^2 \ell_2^2 + \ell_2^2 \ell_3^2 + \ell_3^2 \ell_1^2]) = 0 \tag{55}$$

and

$$\begin{aligned} \sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell &= \sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell \, (\ell_1^2 + \ell_2^2 + \ell_3^2) = \sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell \, (\ell_1^4 + \ell_2^4 + \ell_3^4 - 3\, [\ell_1^2 \ell_2^2 + \ell_2^2 \ell_3^2 + \ell_3^2 \ell_1^2]) \\ &= \sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell \, (\ell_1^4 \ell_2^2 + \ell_1^4 \ell_3^2 + \ell_2^4 \ell_3^2 + \ell_2^4 \ell_1^2 + \ell_3^4 \ell_1^2 + \ell_3^4 \ell_2^2 + \ell_1^2 \ell_2^2 + \ell_2^2 \ell_3^2 + \ell_3^2 \ell_1^2 - 18\, \ell_1^2 \ell_2^2 \ell_3^2) \\ &= \sum_{\ell \in \mathcal{Z}^n_{q-1}} \hat{\gamma}_\ell \, (2\, [\ell_1^6 + \ell_2^6 + \ell_3^6] + 15\, [\ell_1^2 \ell_2^2 + \ell_2^2 \ell_3^2 + \ell_3^2 \ell_1^2] - 90\, \ell_1^2 \ell_2^2 \ell_3^2) = 0 \end{aligned} \tag{56}$$

for the thin plate spline when $n = 2$, the linear and multiquadric functions when $n = 3$, and the cubic radial basis function when $n = 3$ respectively. We see that the objective function (50) and all the constraints of each linear programming problem provide the required symmetry $\{\gamma_{P\ell} = \gamma_\ell \mid P\ell \in \mathcal{P}(\ell)\}$, so we reduce the number of variables of this calculation to the number of equivalence sets $\mathcal{P}(\ell)$ in $\mathcal{Z}^n_q$ or in $\mathcal{Z}^n_{q-1}$, the choice depending on whether we work with $\{\gamma_\ell\}$ or with $\{\hat{\gamma}_\ell\}$. We set $q = 3$ in the $n = 2$ inverse multiquadric case and $q = 4$ otherwise. In all cases we let $a = 6$ or $a = 4$ for $n = 2$ or $n = 3$ respectively. No difficulties occurred in the computations of the quasi-interpolating functions $\psi(\cdot)$ by some software that was developed by one of the authors (MJDP) for the solution of general, linearly constrained, optimization problems.

The purpose of each of the constraints (54), (55) and (56) is to optimize the asymptotic decay properties of $\psi(\cdot)$. Specifically, they ensure that the integer j in the bound

$$|\psi(x)| = O(\|x\|^{-j}), \quad \|x\| \to \infty, \tag{57}$$

is as large as possible, subject to $\psi(\cdot)$ having the form (51) and subject to the *integral condition* that enough freedom remains in the coefficients $\{\hat{\gamma}_\ell \mid \ell \in \mathcal{Z}^n_{q-1}\}$ for it to be possible for $\int_{\mathcal{R}^n} \psi(x)\, dx$ to be nonzero. The details of equations (54)–(56) can be derived from the Taylor series expansion of expression (51) for large $\|x\|$ or from the relations between asymptotic decay rates and Fourier transforms that are mentioned in Section 2. Thus one finds the values $j = 6$, $j = 7$ and $j = 7$ in the bound (57) for the $n = 2$ thin plate spline, the $n = 3$ linear and multiquadric cases and the $n = 3$ cubic radial basis function respectively. Hence the right hand side of the diagonal dominance condition (42) is finite. Further, when q and a have the values given in the previous paragraph, it seems that in each of these cases the $\psi(\cdot)$ from the linear programming calculation actually satisfies inequality (42), provided that c is not too large in the multiquadric radial basis function.

The integral condition that has just been mentioned keeps j finite in inequality (57) by ruling out functions $\psi(\cdot)$ that are formed by applying high order divided difference operators to $\{\phi(\|x\|_2) \mid x \in \mathcal{R}^n\}$, and, more importantly, it is a necessary condition for inequality (42) to hold in each of the cases (9). To justify this statement we deduce a contradiction if we have the diagonal dominance (42), $\psi(\cdot)$ being an absolutely integrable function with $\int_{\mathcal{R}^n} \psi(x)\,dx = 0$. Inequality (42) allows the Gauss–Seidel method, as mentioned in the first paragraph of this section, to construct a function

$$C^{[\infty]}(x) = \sum_{\ell \in \mathcal{Z}^n} \mu_\ell^{[\infty]}\, \psi(x-\ell), \quad x \in \mathcal{R}^n, \tag{58}$$

that satisfies $\{C^{[\infty]}(\ell) = \delta_{0\ell} \mid \ell \in \mathcal{Z}^n\}$, and the sum $\sum_{\ell\in\mathcal{Z}^n} |\mu_\ell^{[\infty]}|$ of moduli of coefficients is finite, which implies $\int_{\mathcal{R}^n} C^{[\infty]}(x)\,dx = 0$. The cardinal functions $C(\cdot)$ that are studied in Section 2 for each of the cases (9), however, have the property that the interpolation formula (5) gives $s \equiv f$ when $f \equiv 1$, which implies $\int_{\mathcal{R}^n} C(x)\,dx = 1$. Therefore $C^{[\infty]}(\cdot)$ is different from $C(\cdot)$. Due to the asymptotic decay properties of $C(\cdot)$ and $C^{[\infty]}(\cdot)$, given in Section 2 and deducible from the smoothness and absolute integrability of $\psi(\cdot)$, we now have a contradiction to the uniqueness result in Theorem 17 of Buhmann (1988b).

It follows from this argument that inequality (42) is satisfied only if $\psi(\cdot)$ is not absolutely integrable or if $\int_{\mathcal{R}^n} \psi(x)\,dx \neq 0$. Remembering that the number of terms of $\hat{\mathcal{Z}}$ in equation (41) is finite, and that $\phi(\cdot)$ is one of the functions (9), the first alternative would imply that the right hand side of expression (42) is a divergent sum. Therefore diagonal dominance occurs only if $\psi(\cdot)$ is absolutely integrable with a nonzero integral. It is proved in Jackson (1988), however, that no such function exists in the linear case $\phi(r) = r$ when n is even. It also follows from his analysis that condition (42) cannot hold for the linear, cubic, multiquadric and inverse multiquadric functions when n is even nor for the thin plate spline when n is odd. Fortunately this lack of diagonal dominance does not prevent satisfactory convergence of the Gauss–Seidel iteration in all the calculations of Section 4.

For example, in the inverse multiquadric case with $c = \frac{1}{2}$ and $n = 2$, the only constraint in the linear programming calculation is the normalization condition (49). With $q = 3$ and $a = 6$, the final value of the objective function (50) is 1.166, which suggests that inequality (42) might hold, but the right hand side of this expression is infinite because $\sum_{\ell\in\mathcal{Z}_3^2} \gamma_\ell = 0.0148$. One can deduce from the bound (47), however, that the Gauss–Seidel iteration converges for values of b up to 15, because the calculated $\psi(\cdot)$ satisfies the condition

$$\sum_{\ell \in \mathcal{Z}_{15}^2 \setminus \{m\}} |\psi(\ell - m)| \leq 0.919, \quad m \in \mathcal{Z}_{15}^2, \tag{59}$$

the inequality being an equation when $m = 0$. In fact the largest value of b in these calculations is $b = 24$, and then $\sum_{\ell\in\mathcal{Z}_b^2\setminus\{0\}} |\psi(\ell)| = 1.810$. Even in this case

Table 1 Some values of $C_b(x)$ when $\phi(r)=r$ and $n=3$

x	$b=4$	$b=5$	$b=6$	$b=7$	$b=8$
(0.5,0,0)	0.399676	0.399676	0.399676	0.399676	0.399676
(1.5,0,0)	-0.021868	-0.021868	-0.021868	-0.021868	-0.021868
(2.5,0,0)	8.7094_{-4}	8.7095_{-4}	8.7095_{-4}	8.7095_{-4}	8.7095_{-4}
(3.5,0,0)	2.7883_{-6}	1.6869_{-6}	1.6721_{-6}	1.6733_{-6}	1.6733_{-6}
(4.5,0,0)	-2.4931_{-5}	-4.8588_{-6}	-5.1298_{-6}	-5.1386_{-6}	-5.1381_{-6}
(5.5,0,0)	-2.6420_{-5}	-6.6890_{-6}	7.3171_{-7}	6.9654_{-7}	6.9279_{-7}
(6.5,0,0)	-6.8362_{-6}	-8.6200_{-6}	-1.3422_{-6}	-5.7800_{-8}	-5.9653_{-8}
(7.5,0,0)	-5.0118_{-7}	-1.0026_{-6}	-1.5452_{-6}	-3.0887_{-7}	-3.3524_{-10}
(8.5,0,0)	1.6935_{-7}	-2.9567_{-8}	-2.3408_{-7}	-3.6663_{-7}	-7.4295_{-8}

the convergence of the Gauss–Seidel iteration is entirely satisfactory, the number of iterations to reduce the maximum residual by a factor of 10 on the 49×49 grid being about 300, where each iteration adjusts at most 8 of the coefficients $\{\mu_\ell^{[k]}\}$ because the set $\mathcal{P}(j)$ contains at most 8 elements. If necessary we could have increased the values of q and a to ensure the convergence of our iterative procedure, but in all cases the original values of q and a give an adequate quasi-interpolating function $\psi(\cdot)$.

The dependence of $C_b(\cdot)$ on b is illustrated in Table 1 for the linear radial basis function in three dimensions, where p_q denotes $p\times 10^q$. In this case $C(\cdot)$ has an exponential asymptotic rate of decay of the form (27), while $\psi(\cdot)$ satisfies condition (57) with $j=7$. Looking across the rows of the table, we see that $C_b(x)$ settles down very quickly as b increases. Therefore much of the last column reflects the decay of the true cardinal function $C(\cdot)$. The later entries of the $b=4$ column, however, show the algebraic rate of decay of $C_b(\cdot)$ that is inherited from $\psi(\cdot)$.

To conclude this section we address the goodness of the approximation $C_b(\cdot)\approx C(\cdot)$, keeping in mind the results of Table 1. We argue that the decay rate of the true cardinal function provides excellent accuracy in $C_b(x)\approx C(x)$ for $0\le \|x\|_\infty \le \frac{1}{2}b$, although $C_b(\cdot)$ may decay much more slowly than $C(\cdot)$. This argument depends on a conjecture, which in it strongest form asserts that, in each of the cases (9), every bounded function has a unique bounded interpolant on $\mathcal{Z}^n$, namely the function (5), where $C(\cdot)$ is the inverse Fourier transform of expression (16). An equivalent claim is that the zero function is the only bounded function in our space of approximations that vanishes on $\mathcal{Z}^n$.

Our argument requires the choice of $\psi(\cdot)$ and the convergence of the Gauss–Seidel iteration to be such that $\{C_b(x) \mid x\in\mathcal{R}^n\}$ is a bounded function, and we also require the test for termination of the iterative procedure to be so fine that we can suppose that $C_b(\cdot)$ satisfies the equations

$$C_b(\ell) = \delta_{0\ell}, \quad \ell\in\mathcal{Z}_b^n. \tag{60}$$

Then the error function

$$\Gamma_b(x) = C(x) - C_b(x), \quad x \in \mathcal{R}^n, \tag{61}$$

is uniformly bounded, and, because it is in the space of approximations, the conjecture of the previous paragraph allows $\Gamma_b(\cdot)$ to be equated to its interpolant, which is the identity

$$\Gamma_b(x) = \sum_{\ell \in \mathcal{Z}^n} \Gamma_b(\ell)\, C(x-\ell), \quad x \in \mathcal{R}^n, \tag{62}$$

the right hand side being absolutely convergent because of the decay properties of $C(\cdot)$ that are given in Section 2. Equations (60)-(62) and $\{C(\ell) = \delta_{0\ell} \mid \ell \in \mathcal{Z}_b^n\}$ imply the bound

$$|\Gamma_b(x)| \leq \|\Gamma_b(\cdot)\|_\infty \sum_{\ell \in \mathcal{Z}^n \setminus \mathcal{Z}_b^n} |C(x-\ell)|, \quad x \in \mathcal{R}^n, \tag{63}$$

and we draw our conclusions from the fact that $\ell \notin \mathcal{Z}_b^n$ on the right hand side. The main points are that only small values of $|C(x-\ell)|$ occur in this absolutely convergent sum when x is well inside the box $\|x\|_\infty \leq b$, and the magnitudes of these small terms depend not on the decay rate of $C_b(\cdot)$ but on the decay rate of $C(\cdot)$. Hence the true cardinal function dominates the approximation $C_b(x) = C(x) - \Gamma_b(x)$ to $C(x)$ when $0 \leq \|x\|_\infty \leq \frac{1}{2}b$, but $\Gamma_b(x)$ becomes important for larger values of $\|x\|_\infty$. For example, $\Gamma_b(\cdot)$ causes the final increases in $|C_b(x)|$ in the $b=7$ column of Table 1. Moreover, because the top half of the $b=8$ column gives excellent approximations to $C(\cdot)$, we see that the contributions from $\Gamma_b(\cdot)$ to the first row of Table 1 are scaled by factors of at most 10^{-5}, which provides the consistency across this row of the table.

In the calculations of the next section the tolerance that stops the Gauss–Seidel iteration is set to such a small number, typically 10^{-16}, that the discrepancies in equation (60) due to $C_b(\cdot) \not\equiv C^{[\infty]}(\cdot)$ are below the rounding errors of the given tables. Further, when estimating each value of $C(x)$ we increased b until the changes to $C_b(x)$ were less than the displayed accuracy. We believe, therefore, that the tabulations of $C(\cdot)$ in Section 4 are close to the true cardinal functions of interpolation, and that there is no need to refer back to the details of the approximations of the algorithm of this section.

4. Numerical results and discussion

Table 2 presents some values of cardinal functions when $n = 2$ in the linear, multiquadric, inverse multiquadric and thin plate spline cases. The lower precision in the penultimate column is due to the rather slow convergence of the sequence of approximations $\{C_b(\cdot) \mid b = 1,2,3,\ldots\}$ when $\phi(r) = (r^2 + \frac{1}{4})^{-1/2}$. From a practical

Table 2 Some values of $C(x)$ when $n=2$

x	$\phi(r)=r$	$(r^2+\frac{1}{4})^{1/2}$	$(r^2+\frac{1}{4})^{-1/2}$	$r^2 \log r$
(0.5,0)	0.434190	0.560007	0.473146	0.535938
(1.5,0)	-0.018506	-0.090066	-0.035897	-0.074666
(2.5,0)	-1.0044_{-3}	0.018726	3.4454_{-3}	0.013094
(3.5,0)	-4.6723_{-5}	-4.0942_{-3}	-1.7455_{-4}	-2.4549_{-3}
(4.5,0)	-2.7134_{-5}	9.2458_{-4}	8.525_{-5}	4.8375_{-4}
(5.5,0)	-1.0850_{-5}	-2.1614_{-4}	3.095_{-5}	-9.8539_{-5}
(6.5,0)	-4.7241_{-6}	5.0800_{-5}	2.114_{-5}	2.0522_{-5}
(7.5,0)	-2.3377_{-6}	-1.2491_{-5}	1.382_{-5}	-4.3399_{-6}
(8.5,0)	-1.2608_{-6}	2.9014_{-6}	9.64_{-6}	9.2796_{-7}
(9.5,0)	-7.2707_{-7}	-7.9513_{-7}	6.97_{-6}	-2.0006_{-7}
(10.5,0)	-4.4266_{-7}	1.4590_{-7}	5.20_{-6}	4.3408_{-8}

point of view the most important question is often 'what is the greatest value of $\|x\|$ such that $|C(x)| \geq 10^{-6}$', because for all larger values of $\|x\|$ one may be able to treat $C(x)$ as negligible. Table 2 shows that, according to this criterion, the thin plate spline is marginally better than the linear and multiquadric radial basis functions, while the inverse multiquadric is least good. If larger values of $|C(x)|$ can be neglected, then $\phi(r)=r$ seems to be the best of the four given cases, but $c=\frac{1}{2}$ is the only c of this tabulation of multiquadric and inverse multiquadric cardinal functions. If one preferred $\phi(r)=(r^2+\frac{1}{100})^{1/2}$, for example, then one would have localization properties that are similar to those of $\phi(r)=r$ and one would have differentiability too.

Because every x in Table 2 is on the first coordinate axis, Figures 1–4 present some properties of the cardinal functions on $\mathcal{R}^2$. Each figure is constructed from the values of $C(\cdot)$ on the 201×201 square grid with mesh size 0.1 on the square $\|x\|_\infty \leq 10$. For each of the heights $\{h=10^{-k} \mid k=1,2,3,\ldots\}$, we calculated the convex hull of the points of this mesh at which $|C(x)| \geq h$. The piecewise linear curves in the figures are the boundaries of these convex hulls in the fourth quadrant, the range of x being $\|x\|_\infty \leq 8$. The shaded regions of the figures are the spaces between the $h=10^{-5}$ and $h=10^{-6}$ curves, which show clearly that the inverse multiquadric radial basis function is the least successful of the four cases.

For each radial basis function $\phi(\cdot)$, the spacings between the convex hull boundaries of the heights $\{h=10^{-k} \mid k=1,2,3,\ldots\}$ indicate the asymptotic decay rate of the cardinal function of interpolation, but, particularly in Figure 2, the range $\|x\|_\infty \leq 8$ is rather small for this purpose. When the cardinal function decays exponentially, which implies a bound of the form (27), then the *average* spacing tends to a constant, and this feature is shown well in Figure 4. In the other three cases, however, the asymptotic decay rate is algebraic as suggested by inequality (25), so the spacings between convex hull boundaries tend to diverge. This property is clear

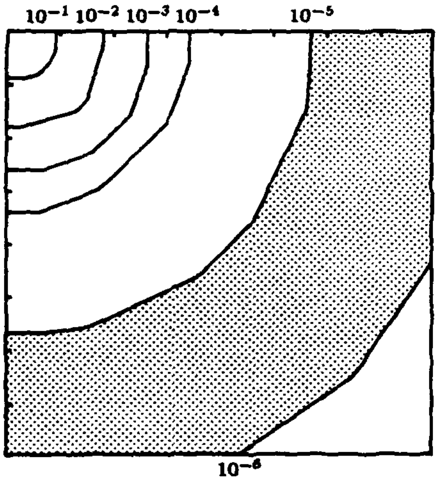


Figure 1

Convex hulls: $\phi(r) = r$

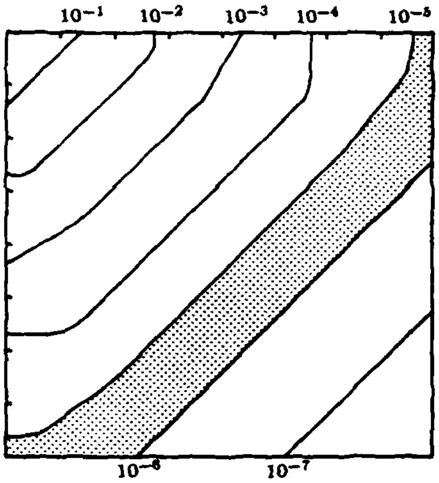


Figure 2

Convex hulls: $\phi(r) = (r^2 + \frac{1}{4})^{1/2}$

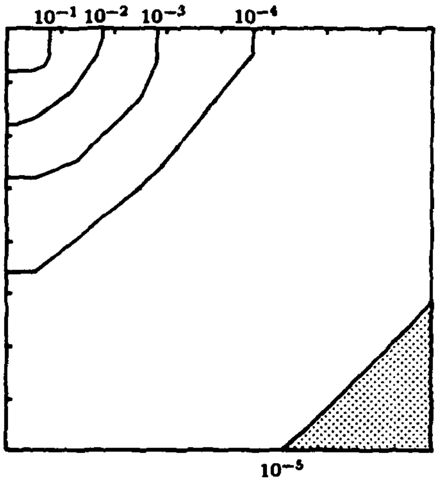


Figure 3

Convex hulls: $\phi(r) = (r^2 + \frac{1}{4})^{-1/2}$

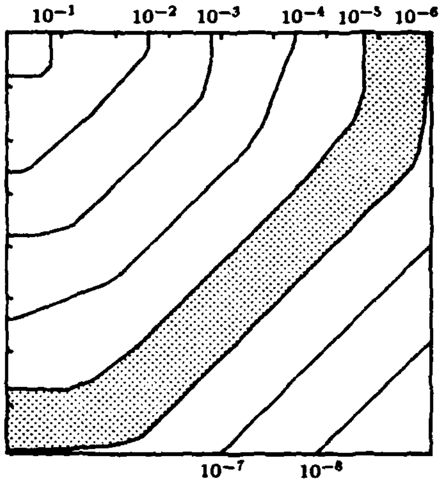


Figure 4

Convex hulls: $\phi(r) = r^2 \log r$

Table 3 Further values of $C(x)$ when $n=2$

	$\phi(r)=(r^2+\frac{1}{4})^{1/2}$		$\phi(r)=r$	
x	$C(x)$	$\|x\|_2^5\, C(x)$	$C(x)$	$\|x\|_2^5\, C(x)$
(10.5,0)	1.4590_{-7}	0.01862	-4.4266_{-7}	-0.05650
(11.5,0)	-6.8753_{-8}	-0.01382	-2.8178_{-7}	-0.05668
(12.5,0)	-4.396_{-9}	-0.00134	-1.8618_{-7}	-0.05682
(13.5,0)	-1.365_{-8}	-0.00612	-1.2698_{-7}	-0.05694
(14.5,0)	-6.928_{-9}	-0.00444	-8.899_{-8}	-0.05704
(15.5,0)	-5.698_{-9}	-0.00510	-6.386_{-8}	-0.05713

in Figures 1 and 3, but the multiquadric figure is more interesting because of its similarity to the exponential decay of the thin plate spline for $h \geq 10^{-6}$.

The shapes of the convex hull boundaries also deserve comment. We know from equation (31) that they tend to be circular in the linear and multiquadric cases, and we see this feature in Figure 1. Asymptotic circularity also occurs for inverse multiquadrics (Buhmann, 1988b), but the range $\|x\|_\infty \leq 8$ is too small for it to appear in Figures 2 and 3. In Figure 4, however, where the asymptotic decay rate of $C(\cdot)$ is exponential, it seems that the convex hull boundaries tend to be diamond shaped ($\|x\|_1 =$ constant). This property is obtained in cubic spline interpolation on $\mathcal{Z}^n$, because the exponential rate of decay along any line that is parallel to a coordinate direction is independent of the displacement of the line from the origin. Perhaps a similar property holds for radial basis function interpolation methods with exponential decay, in which case it would be most appropriate to let $\|x\| = \|x\|_1$ in the bound (27).

The figures so far fail to reflect the asymptotic behaviour of $C(\cdot)$ in the multiquadric case $\phi(r)=(r^2+\frac{1}{4})^{1/2}$. Indeed the signs of the entries in the middle column of Table 2 alternate, but the theory of Section 2, in particular expressions (31) and (38), show that $C(x)$ is negative whenever $\|x\|$ is large and x is away from the points of the grid $\mathcal{Z}^n$. Therefore Table 3 provides some more values of this cardinal function. Now we see that the dominant part of $C(x)$ can be a negative multiple of $\|x\|_2^{-5}$, and that there is fair agreement with the factor -0.005212 in equation (39). It is easier, however, to use tabulated values of $C(\cdot)$ to estimate the constants of expression (32) in the $\phi(r)=r$ case, partly because the first derivative discontinuities of the radial basis function allow $C(x)$ to be nonpositive for all moderate values of $\|x\|$. Thus the last column of Table 3 shows good agreement with the number -0.05747. In order to support the other constant of expression (32), we note that at $x=(9.5,9.5)$, for example, we have $C(x)=-1.54\times 10^{-7}$ and $\|x\|_2^5\, C(x)=-0.0672$.

Some values of cardinal functions in three dimensions are given in Table 4. The asymptotic decay rate of $C(\cdot)$ when $\phi(r)=r$ is particularly impressive, being about a factor of 10 per data point. We recall from the last paragraph of Section 2 that

Table 4 Some values of $C(x)$ when $n=3$

x	$\phi(r)=r$	$(r^2+\frac{1}{100})^{1/2}$	$(r^2+\frac{1}{4})^{1/2}$	$\phi(r)=r^3$
(0.5,0,0)	0.399676	0.439796	0.545232	0.560447
(1.5,0,0)	-0.021868	-0.034425	-0.082773	-0.099110
(2.5,0,0)	8.7095_{-4}	3.2618_{-3}	0.017282	0.025761
(3.5,0,0)	1.6733_{-6}	-3.5220_{-4}	-3.9703_{-3}	7.5461_{-3}
(4.5,0,0)	-5.1381_{-6}	4.3009_{-5}	9.8001_{-4}	2.3896_{-3}
(5.5,0,0)	6.9300_{-7}	-5.7089_{-6}	-2.5541_{-4}	-7.9975_{-4}
(6.5,0,0)	-6.1395_{-8}	7.9934_{-7}	6.9290_{-5}	2.7841_{-4}
(7.5,0,0)	3.5619_{-9}	-1.1589_{-7}	-1.9363_{-5}	-9.9717_{-5}
(8.5,0,0)	-1.9969_{-11}	1.7210_{-8}	5.5319_{-6}	3.6474_{-5}
(9.5,0,0)	-2.8859_{-11}	-2.6008_{-9}	-1.6073_{-6}	-1.3556_{-5}
(10.5,0,0)	4.8041_{-12}	3.9834_{-10}	4.7315_{-7}	5.1017_{-6}

the multiquadric cardinal function decays like $\|x\|_2^{-11}$ as $\|x\| \to \infty$, and we see in the table that this algebraic rate is sufficiently strong for the localization properties of multiquadric interpolation with moderate c to compare favourably with the exponential decay of the $\phi(r)=r^3$ case. Therefore, if one cannot tolerate the first derivative discontinuities in the interpolant when $\phi(r)=r$, it may be better to turn to multiquadric radial basis functions instead of $\phi(r)=r^3$, but the obvious choice for a regular grid is cubic spline interpolation. In large calculations the amount of work of cubic spline methods is orders of magnitude less than that of radial basis function techniques, but perhaps the introduction of transputers and further research will alleviate this gap. Therefore we note that the cubic spline decay rate of $(2-\sqrt{3})$ per data point is less good than the localization properties when $\phi(r)=r$, and that the comparison with multiquadrics depends on c. All of these methods reproduce cubic polynomials in the theoretical case of interpolation on an infinite grid, but only the cubic spline has this property when the grid is finite.

The relevance of approximation on an infinite grid to finite grids has been studied by Dyn and Jackson (private communication, 1988). It is important not only because finiteness is necessary in practical calculations but also because of the gains in efficiency that may be achievable by partitioning very large finite grids into smaller ones. Therefore we give it some attention now. As at the end of Section 3, we relate an interpolant $s_b(\cdot)$ on the finite grid $\mathcal{Z}_b^n \equiv \mathcal{Z}^n \cap [-b,b]^n$ to our interpolant (5) on the infinite grid $\mathcal{Z}^n$. The equation

$$s_b(x) = s(x) - \eta(x), \quad x \in \mathcal{R}^n, \tag{64}$$

defines a function $\eta(\cdot)$ that is in the space of approximations. Therefore, assuming a suitable uniqueness condition for interpolation on $\mathcal{Z}^n$, we may equate $\eta(\cdot)$ to its interpolant, which yields the identity

$$s(x)-s_b(x)=\sum_{\ell\in\mathcal{Z}^n\backslash\mathcal{Z}_b^n}\{f(\ell)-s_b(\ell)\}\,C(x-\ell),\quad x\in\mathcal{R}^n. \tag{65}$$

Even if $f(\cdot)$ is bounded, the magnitude of $s_b(\ell)$ for large $\|\ell\|$ depends on the growth of $\phi(r)$ as $r\rightarrow\infty$. Thus, corresponding to the derivation of inequality (63) from equation (62), we find a bound of the form

$$|s(x)-s_b(x)|\leq\beta\sum_{\ell\in\mathcal{Z}^n\backslash\mathcal{Z}_b^n}\{1+\phi(\|\ell\|_2)\}\,|C(x-\ell)|,\quad x\in\mathcal{R}^n, \tag{66}$$

where β is a constant, the sum being absolutely convergent for all of the cardinal functions that are considered in Section 2. Again all the factors $|C(x-\ell)|$ are small when x is well inside the box $\|x\|_\infty\leq b$. Hence the extent to which $s_b(x)$ enjoys the properties of $s(x)$ depends on the distance from x to the set $\mathcal{Z}^n\backslash\mathcal{Z}_n^b$. If $n=3$ and $\phi(r)=r$, for example, Table 4 suggests the strong result that $|s(x)-s_b(x)|$ is $O(10^{-d})$, where $d=\min\{\|x-\ell\|_\infty\mid\ell\in\mathcal{Z}^n\backslash\mathcal{Z}_b^n\}$, so there is rapid decay in the perturbations to the approximation from the finiteness of the grid.

Tables 2–4 give excellent support to the assumptions on uniqueness of interpolation that are crucial to equations (62) and (65). Here we have in mind that, for each of the radial basis functions (9), the inverse Fourier transform of the function (16) provides a unique cardinal function of the form (6) whose coefficients $\{c_\ell\mid\ell\in\mathcal{Z}^n\}$ satisfy the asymptotic conditions of Theorem 17 of Buhmann (1988b). These conditions correspond to the decay of $|C(x)|$ as $\|x\|\rightarrow\infty$ that is studied in Section 2. If, however, the equations $\{C(\ell)=\delta_{0\ell}\mid\ell\in\mathcal{Z}^n\}$ failed to define a unique bounded $C(\cdot)$ of the form (6), then different attempts to calculate $C(\cdot)$ might yield different cardinal functions. Now the method of the algorithm of Section 3 has nothing to do with the inversion of Fourier transforms, and we even apply the Gauss–Seidel iteration in some cases when the diagonal dominance (42) is not obtained. Thus it is likely that our algorithm would find a cardinal function that is different from the inverse transform of $\hat{C}(\cdot)$ if it were possible to do so. Tables 2–4, however, seem to confirm the asymptotic properties that are established in Section 2. Further, when $\psi(\cdot)$ satisfies the diagonal dominance condition (42), the proof of Lemma 10 of Buhmann (1988a) shows that the Gauss–Seidel iteration of Section 3 gives the cardinal function that is the inverse Fourier transform of $\hat{C}(\cdot)$. Therefore we believe that the uniqueness assumptions are true.

Although the algorithm of Section 3 is suitable for calculating the cardinal functions that we have studied, it is not recommended for interpolation to a general function on a finite grid. Our reasons include the loss of equivalence symmetry and the need for a range of quasi-interpolating functions $\psi(\cdot)$ in order to take account of the effects of edges of the grid. Therefore some other algorithms are currently under investigation at Cambridge, particularly the use of preconditioned conjugate gradient methods for solving the interpolation equations. Such methods have already been applied successfully by Dyn, Levin and Rippa (1986) when the data are not confined to a regular grid.

In conclusion we note that the asymptotic properties of cardinal functions are so encouraging that many more questions deserve attention. Here are three examples. If we estimate the interpolant $s(x)$ in a way that ignores the function values $\{f(\ell) \mid \ell \in \mathcal{Z}^n\}$ for large $\|\ell - x\|$, is there a good way of making the estimate a continuous function of x? If we extend the space of approximating functions to include low order polynomials explicitly, can the interpolant on $\mathcal{Z}_b^n$ provide not only reproduction of these low order polynomials but also the localization properties that are noted in the paragraph that includes expressions (64)–(66)? Although the analysis of Section 2 is an application of Fourier techniques, do the main conclusions of this section require interpolation on a *regular* grid? Clearly there is much practical and theoretical work to be done, and it seems probable that it will yield some very powerful new algorithms for multivariable approximation.

References

Abramowitz, M. and Stegun, I.A. (1970), *Handbook of Mathematical Functions*, Dover Publications (New York).

Buhmann, M.D. (1988a), "Multivariate interpolation in odd dimensional Euclidean spaces using multiquadrics", Technical Report DAMTP 1988/NA6, University of Cambridge.

Buhmann, M.D. (1988b), "Multivariate interpolation with radial basis functions", Technical Report DAMTP 1988/NA8, University of Cambridge.

Dyn, N., Levin, D. and Rippa, S. (1986), "Numerical procedures for surface fitting of scattered data by radial functions", *SIAM J. Sci. Stat. Comput.* **7**, 639–659.

Jackson, I.R.H. (1987), "An order of convergence for radial basis functions", Technical Report DAMTP 1987/NA11, University of Cambridge.

Jackson, I.R.H. (1988), "Convergence properties of radial basis functions", *Constructive Approximation* **4**, 243–264.

Jones, D.S. (1982), *The Theory of Generalised Functions*, Cambridge University Press (Cambridge).

Madych, W.R. and Nelson, S.A. (1987), "Polyharmonic cardinal splines", preprint.

Micchelli, C.A. (1986), "Interpolation of scattered data: distance matrices and conditionally positive functions", *Constructive Approximation* **2**, 11–22.

Powell, M.J.D. (1981), *Approximation Theory and Methods*, Cambridge University Press (Cambridge).

Powell, M.J.D. (1987), "Radial basis functions for multivariable interpolation: a review", in *Algorithms for Approximation*, eds. Mason, J.C. and Cox, M.G., Oxford University Press (Oxford), pp. 143–167.

Stein, E.M. and Weiss, G. (1971), *Introduction to Fourier Analysis on Euclidean Spaces*, Princeton University Press (Princeton).

THE FOURIER OPERATOR OF EVEN ORDER AND ITS APPLICATION TO AN EXTREMUM PROBLEM IN INTERPOLATION

L. BRUTMAN
Department of Mathematics and Computer Science
University of Haifa

Abstract The Fourier operator of even order $\tilde{S}_n$ is defined as a projection of the family of continuous 2π-antiperiodic functions onto the trigonometric polynomials of half-angles. It is shown that this operator is a natural extension of the classical Fourier operator and corresponds to the case where the dimension of the projection subspace is even. The operator $\tilde{S}_n$ is used to extend a result of Szabados, concerning optimal choice of nodes for trigonometric interpolation, to the case of an even number of nodes as well as to the case of complex interpolation by algebraic polynomials on the unit circle.

Key words: Fourier operator, Minimum norm projection, Trigonometric interpolation.

1. The Fourier operator of even order

Various problems of approximation theory involve the trigonometric polynomials of half-angles

$$\tilde{h}_n(\theta) = \sum_{k=1}^{n} [c_k \cos(2k-1)\frac{\theta}{2} + d_k \sin(2k-1)\frac{\theta}{2}]$$

Properties of these polynomials and their application to interpolation and to the theory of quadrature were considered by Tureckiĭ (1960,1968). Recently (see Brutman and Pinkus (1980)) the polynomials $\tilde{h}_n(\theta)$ were found to be useful in proving the Erdös conjecture concerning minimal norm interpolation on the unit circle for even number of points. In the following the trigonometric polynomials of half-angles are used in order to define an even-order analogue of the classical Fourier operator.

Let $\tilde{C}_{2\pi}$ be the Banach space of functions continuous and antiperiodic on $[0,2\pi)$ (i.e. satisfying $f(0) = -f(2\pi)$), equipped with the uniform norm, and denote by $\tilde{H}_n$ the following 2n-dimensional subspace of $\tilde{C}_{2\pi}$

$$\tilde{H}_n = \text{span}\{\sin\frac{\theta}{2}, \cos\frac{\theta}{2}, \ldots, \sin\frac{(2n-1)\theta}{2}, \cos\frac{(2n-1)\theta}{2}\}$$

It is known that $\tilde{H}_n$ is an orthogonal system on $[0,2\pi)$. The Fourier-type operator $\tilde{S}_n : \tilde{C}_{2\pi} \to \tilde{H}_n$ is defined by

$$(\tilde{S}_n f)(\theta) = \sum_{k=1}^{n} \{c_k \cos(2k-1)\frac{\theta}{2} + d_k \sin(2k-1)\frac{\theta}{2}\}, \qquad (1)$$

where

$$c_k = \frac{1}{\pi}\int_0^{2\pi} f(\theta)\cos(2k-1)\frac{\theta}{2}\, d\theta,$$
$$d_k = \frac{1}{\pi}\int_0^{2\pi} f(\theta)\sin(2k-1)\frac{\theta}{2}\, d\theta. \qquad (2)$$

The following integral representation of $\tilde{S}_n$ holds

Theorem 1

$$(\tilde{S}_n f)(\theta) = \frac{1}{\pi}\int_0^{2\pi} f(\tau)\tilde{D}_n(\tau-\theta)d\tau, \tag{3}$$

where

$$\tilde{D}_n(\theta) = \sum_{k=1}^{n} \cos(2k-1)\frac{\theta}{2} = \frac{\sin n\theta}{2\sin(\theta/2)} . \tag{4}$$

Note that the kernel $\tilde{D}_n(\theta)$ in (3) is a natural extension of the classical Dirichlet kernel $D_m(\theta) = \sin(m\theta/2)/2\sin(\theta/2)$, $m = 2n+1$, and corresponds to the case of even m. As a direct consequence of the representation (3) we get

Corollary 1

$$\|\tilde{S}_n\| = \frac{2}{\pi}\int_0^{\pi/2} \frac{|\sin 2n\theta|}{\sin\theta} d\theta \equiv \tilde{\rho}_n . \tag{5}$$

Remark To the best of our knowledge, the quantities $\tilde{\rho}_n$ appear for the first time in Szegö (1921): "Mann kann die Konstanten

$$r_m = \frac{2}{\pi}\int_0^{\pi/2} \frac{|\sin m\theta|}{\sin} d\theta , \quad m = 0,1,2,\dots$$

betrachten, welche sich für ungerade m auf die Lebesgueschen Konstanten reduzieren." Szegö has proved the following representation

$$r_m = \frac{16}{\pi^2}\sum_{\nu=1}^{\infty} \left[\frac{1}{1} + \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{2m\nu-1}\right]\frac{1}{4\nu^2-1} , \tag{6}$$

which was used by Galkin (1971) to obtain the precise estimate

$$0.9894 < r_m - \frac{4}{\pi^2}\log m \leq 1. \tag{7}$$

The quantities $\tilde{\rho}_n$ were also used by Geddes and Mason (1975) as an upper bound for the norm of the Taylor projection of even order. They gave the following formula

$$\tilde{\rho}_n = \frac{4}{\pi}\sum_{k=1}^{n}\frac{1}{2k-1}\tan\frac{2k-1}{4n}\pi \qquad (8)$$

which is analogous to the representation of the classical Lebesgue constants due to Fejér (1910).

Continuing the analogy with the classical case we define the Fejer-type operator $\tilde{F}_n : \tilde{C}_{2\pi} \to \tilde{H}_n$ to be the arithmetic mean:

$$\tilde{F}_n = \frac{1}{n}\{\tilde{S}_1 + \tilde{S}_2 + \ldots + \tilde{S}_n\} \qquad (9)$$

Then we have

Theorem 2

$$(\tilde{F}_n f)(\theta) = \frac{1}{n\pi}\int_0^{2\pi} f(\tau)\tilde{\sigma}_n(\tau-\theta)d\tau, \qquad (10)$$

where

$$\tilde{\sigma}_n(\theta) = \frac{\sin\frac{n+1}{2}\theta\,\sin\frac{n\,\theta}{2}}{2\sin^2\frac{\theta}{2}} \quad . \qquad (11)$$

Note that in contrast to the classical case, the kernel $\tilde{\sigma}_n(\theta)$ is not positive. Yet this is not surprising, since the operator $\tilde{F}_n$ is defined on the space $\tilde{C}_{2\pi}$ which does not contain positive functions. This observation raises the natural question of whether every $f \in \tilde{C}_{2\pi}$ can be uniformly approximated by the Fejér-type polynomials $\tilde{F}_n(\theta)$. It can be shown by applying the classical result of Fejér (1904) that the answer to this question is affirmative, namely

Theorem 3 The set $\tilde{H} = \{\cos(\theta/2), \sin(\theta/2), \cos(3\theta/2), \sin(3\theta/2), \ldots\}$ is dense in $\tilde{C}_{2\pi}$.

Applying Theorem 3 and arguments analogous to those used for the classical trigonometric polynomials, the following analog of the Marcinkiewicz-Zygmund-Berman identity can be proven.

Theorem 4 Let $\tilde{P}_n$ be any projection from $\tilde{C}_{2\pi}$ onto $\tilde{H}_n$. Then

$$\frac{1}{2\pi}\int_0^{2\pi}\tilde{P}_n(f(x-\tau),\theta+\tau)d\tau = (\tilde{S}_n f)(\theta) \qquad (12)$$

As a consequence of the above theorem we have the following minimal norm property of $\tilde{S}_n$ (For the corresponding classical results see Losinskiĭ (1948) and Cheney et al. (1969)).

Corollary 2

$$||\tilde{S}_n|| \le ||\tilde{P}_n|| \tag{13}$$

Moreover, the equality in (13) holds only if $\tilde{P}_n = \tilde{S}_n$.

2. An interpolation-theoretical extremum problem.

Tureckii (1968) posed the following problem: Let $0 \le \theta_0 < \theta_1 < \dots < \theta_{2n} < 2\pi$ and let $t_k(\theta) = t_{k,n}(\theta)$, $k=0,.,\dots,2n$ be the fundamental trigonometric polynomials of degree n, i.e. such that $t_k(\theta_j) = \delta_{k,j}$, $k,j=0,1,\dots,2n$. For what system of nodes $\{\theta_k\}_{k=0}^{2n}$ will

$$I_p = I_p(\theta_0,\theta_1,\dots,\theta_{2n}) = \int_0^{2\pi} \{ \sum_{k=0}^{2n} |t_k(\theta)|^p \} d\theta, \quad (0<p<\infty) \quad (14)$$

be minimal? It was conjectured by Tureckiĭ that the optimal solution corresponds to the equidistant nodes. Schumacher (1976) proved this conjecture in case p=1, as a corollary of a much more general theorem. Szabados (1980) found a direct solution to the problem in the case $1 \le p < \infty$, specifically he proved the following

Theorem 5 Let $p \ge 1$. The integral $I_p(\theta_0,\theta_1,\dots,\theta_{2n})$ is minimal iff the nodes are identical with the equidistant nodes $\theta_k = 2k\pi/(2n+1)$, $k=0,1,\dots,2n$, or with their translation. Moreover,

$$\min_{\theta_k} I_p(\theta_0,\theta_1,\dots,\theta_{2n}) = \frac{2^p}{(2n+1)^{p-1}} \int_0^{\pi} |D_n(\theta)|^p d\theta \tag{15}$$

where $D_n(\theta)$ is the classical Dirichlet kernel.

In the following we extend the above result to the case where the number of interpolation nodes is even. Let $0 \le \theta_1 < \theta_2 < \dots < \theta_{2n} < 2\pi$ and denote by $\tilde{t}_k(\theta) = \tilde{t}_{k,n}(\theta)$, $k=1,2,\dots,2n$, the fundamental trigonometric polynomials of half-angles, i.e. $\tilde{t}_k \in \tilde{H}_n$ and $\tilde{t}_k(\theta_j) = \delta_{kj}$. Let

$$\tilde{I}_p = \tilde{I}_p(\theta_1,\theta_2,\dots,\theta_{2n}) = \int_0^{2\pi}\{\sum_{k=1}^{2n}|\tilde{t}_k(\theta)|^p\}d\theta \tag{16}$$

The following result holds

Theorem 6 The integral $\tilde{I}_p(\theta_1,\theta_2,\dots,\theta_{2n})$, $p \geq 1$, is minimal iff the nodes are identical with $\theta_k = k\pi/n$, $k=0,1,\dots,2n-1$, or with their translation. Moreover

$$\min_{\theta_k} \tilde{I}_p(\theta_1,\theta_2,\dots,\theta_{2n}) = \frac{2^p}{(2n)^{p-1}}\int_0^{2\pi}|\tilde{D}_n(\theta)|^p d\theta, \tag{17}$$

where $\tilde{D}_n(\theta)$ is the Dirichlet kernel corresonding to the trigonometric polynomials of half-angles. The proof of this theorem, which is based on (12), follows the same reasoning as the proof of Szabados.

Our final result concerns complex interpolation by algebraic polynomials on the unit circle. Let $z_k=\exp(i\theta_k)$, $k=0,1,\dots,m$ be $m+1$ distinct points and let $l_{k,m}(z)$, $k=0,.,\dots,m$ be the corresponding fundamental polynomials. It was proved by Brutman (1980) that

$$\begin{aligned} |\ell_{k,m}(e^{i\theta})| &= |t_{k,n}(\theta)|, \quad m=2n \\ &= |\tilde{t}_{k,n}(\theta)|, \quad m=2n-1 \end{aligned} \tag{18}$$

Combining (18) with Theorems 4 and 5, we arrive at the following

Theorem 7 The integral

$$I_p(z_0,z_1,\dots,z_m) = \int_0^{2\pi}\{\sum_{k=0}^{m}|\ell_{k,m}(\ell^{i\theta})|^p\}d\theta,\ p \geq 1 \tag{19}$$

is minimal iff $z_k = \exp[ik\pi/(m+1)+\alpha]$, $k=0,1,\dots,m$, some α. Moreover

$$\min_{z_k} I_p(z_0,z_1,\dots,z_m) = \frac{2^p}{(m+1)^{p-1}}\int_0^{2\pi}|D_m(\theta)|^p d\theta \tag{20}$$

where $D_m(\theta)=\sin(m\theta/2)/2\sin(\theta/2)$ is a generalized Dirichlet kernel.

References

Brutman, L. (1980), On the polynomial and rational projections in the complex plane, SIAM J. Numer. Anal. 17, 366-372

Brutman, L. and Pinkus, A. (1980), On the Erdös conjecture concerning minimal norm interpolation on the unit circle, SIAM J. Numer. Anal. 17, 373-375

Cheney, C.W., Hobby, C.R., Morris, P.D., Shurer, F. Wulbert, D.E. (1969) On the minimal property of the Fourier projection, Trans. Amer. Math. Soc. 143, 249-258

Fejér, L. (1904), Untersuchungen über Fouriersche Reihen, Math. Ann. 58, 51-69.

Fejér, L. (1910), Lebesguesche Konstanten und divergente Fourierreihen, J. Reine Angew. Math. 138, 22-53

Galkin, P.V. (1971), Estimates for the Lebesgue constants, Proc. Steklov Inst. Math. 109, 1-3

Geddes, K.O. and Mason, J.C. (1975), Polynomial approximation by projections on the unit circle, SIAM J. Numer. Anal. 1, 111-120

Lozinskiĭ, S.M. (1948), On a class of linear operators, Dokl. Akad. Nauk SSSR 61, 193-196

Schumacher, R. (1976), Zur Minimalität trigonometrischer Polynomoperatoren, Manuscripta Math. 19, 133-142

Szabados, J. (1980), On an interpolation-theoretic extremum problem, Periodica Math. Hungarica 11, 145-150

Szegö, G. (1921), Über die Lebesgueschen Konstanten bei den Fourierschen Reihen, Math. Z. 9, 163-166.

Turečkiĭ, A.H. (1960), On the quadrature formulas with even number of nodes accurate for the trigonometric polynomials, Dokl. Akad. Nauk BSSR 9, 365-366 (In Russian)

Turečkiĭ, A.H. (1968), 'Theory of Interpolation in Problem Form', Vyšeišaya Škola, Minsk, (In Russian); MR 41, #5840

ON MULTIVARIATE POLYNOMIAL INTERPOLATION

N. DYN[†], A. RON[*]

[†]School of Mathematical Sciences, Tel Aviv University

[*]Department of Computer Science, University of Wisconsin-Madison

Abstract A class of spaces of multivariate polynomials, closed under differentiation, is studied and corresponding classes of well posed Hermite-type interpolation problems are presented. All Hermite-type problems are limits of well posed Lagrange problems.

The results are based on a duality between certain spaces of multivariate exponential-polynomials $\mathcal{H}$ and corresponding spaces of multivariate polynomials $\mathcal{P}$, used by Dyn and Ron (1988) to establish the approximation order of the span of translates of exponential box splines. In the interpolation theory $\mathcal{P}$ is the space of interpolating polynomials and $\mathcal{H}$ characterizes the interpolation points and the interpolation conditions, both spaces being defined in terms of a set of hyperplanes in $\mathbb{R}^s$.

This geometric approach extends the work of Chung and Yao (1977) on Lagrange interpolation, and also a subset of the Hermite-type problems considered via the Newton scheme, by several authors (see Gasca and Maetzu (1982) and references therein). For a different approach to the interpolation problem see Chui and Lai (1988).

It is the systematic and unified analysis of a wide class of interpolation problems which is the main contribution of this paper to the study of multivariate polynomial interpolation.

Keywords: Multivariate interpolation, multivariate polynomials, Hermite-type interpolation.

1. The Interpolating Polynomial Spaces

The spaces of interpolating polynomials we consider here are more general than the total degree polynomials π_m (polynomials of degree $\leq m$), and are still closed under differentiation.

Given a set of directions $A = \{\mathbf{a}^1, \ldots, \mathbf{a}^n\} \subset I\!R^s$, with the property $\operatorname{span} A = I\!R^s$, consider the space of polynomials

$$\mathcal{P}(A) = \operatorname{span}\left\{ \prod_{i \in I} (\mathbf{a}^i \cdot \mathbf{x}) \,\middle|\, I \in S(A) \right\} \tag{1}$$

where $S(A)$ consists of index sets corresponding to "small enough" subsets of A, namely

$$S(A) = \left\{ I \subset \{1, \ldots, n\} \mid \operatorname{span}\{\mathbf{a}^i \mid i \notin I\} = I\!R^s \right\} . \tag{2}$$

By choosing $I \in S(A)$ such that $\{1, \ldots, n\} \backslash I$ is a basis of $I\!R^s$, we conclude that

$$\mathcal{P}(A) \subset \pi_{n-s} . \tag{3}$$

To see that $\mathcal{P}(A)$ is closed under differentiation, observe that

$$\frac{\partial}{\partial x_j} \prod_{i \in I} (\mathbf{a}^i \cdot \mathbf{x}) = \sum_{\ell \in I} a_j^\ell \prod_{\substack{i \in I \\ i \neq \ell}} (\mathbf{a}^i \cdot \mathbf{x}) , \tag{4}$$

and that if $I \in S(A)$ then any subset of I is in $S(A)$.

A more involved analysis is required in order to show the following two properties of $\mathcal{P}(A)$, demonstrated in Dyn and Ron (1988):

(a) Let $d = d(A) = \min\left\{ |I| \mid I \subset \{1, \ldots, n\} \text{ , } I \notin S(A) \right\}$. Then $\pi_{d-1} \subset \mathcal{P}(A)$.

(b) The dimension of $\mathcal{P}(A)$ equals the number of bases that can be formed from A.

Combining (a) and (3) we conclude that

$$\pi_{d-1} \subset \mathcal{P}(A) \subset \pi_{n-s} . \tag{5}$$

If $\mathbf{a}^1, \ldots, \mathbf{a}^n$ are in "general position", namely any s vectors among $\mathbf{a}^1, \ldots, \mathbf{a}^n$ form a basis of $I\!R^s$, then it is easy to see that $d = n - s + 1$. Hence $\mathcal{P}(A) = \pi_{n-s}$.

To introduce a basis of $\mathcal{P}(A)$, consider n hyperplanes

$$H_i = \{x \in I\!R^s \mid \mathbf{a}^i \cdot \mathbf{x} = \gamma_i\} , \qquad i = 1, \ldots, n , \tag{6}$$

determined by $\Gamma = (\gamma_1, \ldots, \gamma_n) \in I\!R^n$. For each $\mathbf{v} \in I\!R^s$ define

$$I_{\mathbf{v}} = \left\{ i \in \{1, \ldots, n\} \mid \mathbf{v} \in H_i \right\} . \tag{7}$$

and consider the set of intersection points of $H_1, \ldots, H_n$,

$$V(A,\Gamma) = \{\mathbf{v} \in I\!R^s \mid \operatorname{span} A_{\mathbf{v}} = I\!R^s\} \ , \qquad A_{\mathbf{v}} = \{\mathbf{a}^i \mid i \in I_{\mathbf{v}}\} \ . \tag{8}$$

Choosing Γ so that $|I_{\mathbf{v}}| = s$ for $\mathbf{v} \in V(A,\Gamma)$, where $|I_{\mathbf{v}}|$ denotes the cardinality of $I_{\mathbf{v}}$, we conclude from (b) that

$$\dim \mathcal{P}(A) = |V(A,\Gamma)| \ . \tag{9}$$

Furthermore, the following polynomials

$$p_{\mathbf{v}}(\mathbf{x}) = \prod_{i \notin I_{\mathbf{v}}} \frac{(\mathbf{a}^i \cdot \mathbf{x} - \gamma_i)}{(\mathbf{a}^i \cdot \mathbf{v} - \gamma_i)} \ , \qquad \mathbf{v} \in V(A,\Gamma) \ , \tag{10}$$

are linearly independent, since

$$p_{\mathbf{v}}(\mathbf{u}) = \begin{cases} 0 & \mathbf{u} \neq \mathbf{v} \ , \\ 1 & \mathbf{u} = \mathbf{v} \ . \end{cases} \qquad \mathbf{v}, \mathbf{u} \in V(A,\Gamma) \ , \tag{11}$$

and hence constitute a basis of $\mathcal{P}(A)$.

The pair (A,Γ) is termed "simple" (for simple intersection points as opposed to multiple ones) if $|I_{\mathbf{v}}| = s$ for all $\mathbf{v} \in V(A,\Gamma)$.

Remark 1 It is shown by Ron (1988) that for fixed A the set of all $\Gamma \in I\!R^n$ such that (A,Γ) is simple, is dense in $I\!R^n$.

The explicit form (10) of a basis of $\mathcal{P}(A)$ indicates that the following result holds.

Proposition 1 $\mathcal{P}(A)$ consists of polynomials of degree $\leq n - s$, which are of degree $\leq n - s - \big|\{i \in \{1,\ldots,n\} \mid \mathbf{a}^i \in \operatorname{span}\{\mathbf{y}\}\}\big| + 1$ along hyperplanes of the form $\mathbf{y} \cdot \mathbf{x} = \lambda$, $\mathbf{y} \in A$, $\lambda \in I\!R$. Furthermore, let $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^k\}$, be $k < s$ pairwise distinct directions in A. Then the degree of any $p \in \mathcal{P}(A)$ along the intersection of k hyperplanes of the form

$$\mathbf{y}^j \cdot \mathbf{x} = \mu_j \ , \quad j = 1, \ldots, k \ , \qquad \mu_1, \ldots, \mu_k \in R \ , \tag{12}$$

is at most

$$n - s - \big|\{i \in \{1,\ldots,n\} \mid \mathbf{a}^i \in \langle Y \rangle\}\big| + \dim \langle Y \rangle \ , \tag{13}$$

where $\langle Y \rangle = \operatorname{span} Y$.

Proof Since for (A, Γ) simple, and $\mathbf{v} \in V(A, \Gamma)$, $\{\mathbf{a}^i \mid i \in I_{\mathbf{v}}\}$ is a basis of $I\!R^s$, each $p_{\mathbf{v}}$ in (10) consists of at least $\left|\{i \in \{1, \dots, n\} \mid \mathbf{a}^i \in \operatorname{span}\{\mathbf{y}\}\}\right| - 1$ factors which are constant along $\mathbf{y} \cdot \mathbf{x} = \lambda$, $\lambda \in I\!R$. Similarly, one can count the constant factors in $p_{\mathbf{v}}$ of (10) along the intersection of the hyperplanes (12), to conclude (13).

Remark 2 The space $\mathcal{P}(A)$ consists of all polynomials over $I\!R^s$ with the properties stated in Proposition 1. This will be shown elsewhere.

2. The Interpolation Problems

In this section we present a class of interpolation problems which are unisolvent in $\mathcal{P}(A)$ for fixed A. The interpolation points and the data at each point, which is of Hermite type, are determined by the choice of $\Gamma = (\gamma_1, \dots, \gamma_n) \in I\!R^n$. The set of interpolation points consists of all points of intersection of at least s of the hyperplanes (6), namely, it is the set denoted by $V(A, \Gamma)$. To define the interpolation conditions at each $\mathbf{v} \in V(A, \Gamma)$, we consider the set of *directions related to* $\mathbf{v}$

$$A_{\mathbf{v}} = \{\mathbf{a}^i \mid i \in I_{\mathbf{v}}\} , \tag{14}$$

and a corresponding polynomial space defined by

$$\mathcal{K}(A_{\mathbf{v}}) = \{p \in \pi \mid [\prod_{i \in I} (\mathbf{a}^i \cdot D)] p \equiv 0 \ , \ I \notin S(A_{\mathbf{v}})\} , \tag{15}$$

where $D = (\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_s})$. Since each I in (15) satisfies $|I| \geq d_{\mathbf{v}} = d(A_{\mathbf{v}})$, it is clear that $\pi_{d_{\mathbf{v}}-1} \subset \mathcal{K}(A_{\mathbf{v}})$. The space $\mathcal{K}(A_{\mathbf{v}})$ is closed under differentiation since $D^{\mathbf{m}}$ commutes with any polynomial in D. In terms of $\mathcal{K}(A_{\mathbf{v}})$ the interpolation conditions at $\mathbf{v}$ are

$$[q(D)p](\mathbf{v}) = [q(D)f](\mathbf{v}) \ , \qquad q \in \mathcal{K}(A_{\mathbf{v}}) \ , \tag{16}$$

where f is smooth enough, and $q(D)$ is obtained from the polynomial $q(\mathbf{x})$ by replacing the vector $\mathbf{x}$ by the vector D. With these definitions we can introduce the interpolation problem determined by A and Γ:

$$\text{Find } p \in \mathcal{P}(A) \text{ satisfying (16) for all } \mathbf{v} \in V(A, \Gamma). \tag{17}$$

The solvability of (17) is due to the following result from Dyn and Ron (1988):

Theorem 1 The spaces $\mathcal{P}(A)$ and the space

$$\mathcal{H}(A,\Gamma) = \bigoplus_{\mathbf{v}\in V(A,\Gamma)} \left\{ e^{\mathbf{v}\cdot\mathbf{x}} q(\mathbf{x}) \mid q \in \mathcal{K}(A_\mathbf{v}) \right\} , \tag{18}$$

are dual to each other under the pairing

$$[p(D)h](0) = [q(D)p](\mathbf{v}) , \quad p \in \mathcal{P}(A) , \quad h(\mathbf{x}) = e^{\mathbf{v}\cdot\mathbf{x}} q(\mathbf{x}) \in \mathcal{H}(A,\Gamma) . \tag{19}$$

Corollary 1 There exists a unique $p \in \mathcal{P}(A)$ solving the interpolation problem (17).

It follows from Theorem 1 that $\mathcal{K}(A_\mathbf{v})$ is dual to $\mathcal{P}(A_\mathbf{v})$ in the sense of (19), and by (b) $\dim \mathcal{K}(A_\mathbf{v}) = \#$ of bases in $A_\mathbf{v}$. Furthermore, since $\mathcal{P}(A_\mathbf{v}) \subset \pi_{|A_\mathbf{v}|-s}$ we conclude that $\mathcal{K}(A_\mathbf{v}) \subset \pi_{|A_\mathbf{v}|-s}$. Hence

$$\pi_{d_\mathbf{v}-1} \subset \mathcal{K}(A_\mathbf{v}) \subset \pi_{|A_\mathbf{v}|-s} , \tag{20}$$

in analogy to (5). Moreover, if the directions in $A_\mathbf{v}$ are in general position then $\mathcal{K}(A_\mathbf{v}) = \pi_{|A_\mathbf{v}|-s}$.

Corollary 2 Let Γ be such that for each $\mathbf{v} \in V(A,\Gamma)$ the directions in $A_\mathbf{v}$ are in general position. Then the interpolation conditions in (16) are pure Hermite of the form

$$D^{\mathbf{m}} p(\mathbf{v}) = D^{\mathbf{m}} f(\mathbf{v}) , \qquad |\mathbf{m}| = \sum_{i=1}^{s} m_i \le |A_\mathbf{v}| - s , \ m_i \ge 0 \ , \ i = 1,\ldots,s . \tag{21}$$

In case A consists of directions in general position, then so does each $A_\mathbf{v}$, $\mathbf{v} \in V(A,\Gamma)$, and the interpolation problem becomes: Find $p \in \pi_{n-s}$ satisfying (21) for each $\mathbf{v} \in V(A,\Gamma)$. In R^2 the conditions on Γ in Corollary 2 are satisfied if $\gamma_i \neq \lambda\gamma_j$ whenever $\mathbf{a}^i = \lambda\mathbf{a}^j$, $\lambda \in \mathbb{R}$, $i \neq j$, $i,j \in \{1,\ldots,n\}$, namely if the hyperplanes $H_1,\ldots,H_n$ in (6) are pairwise disjoint.

An especially interesting interpolation problem is the Lagrange interpolation, obtained when (A,Γ) is simple. In this case $|A_\mathbf{v}| = s$, $\mathcal{K}(A_\mathbf{v}) = \pi_0$, and p satisfies $p(\mathbf{v}) = f(\mathbf{v})$, $\mathbf{v} \in V(A,\Gamma)$. The solution is given explicitly, in terms of the basis (10), as

$$p(x) = \sum_{\mathbf{v}\in V(A,\Gamma)} f(\mathbf{v}) p_\mathbf{v}(x) . \tag{22}$$

This together with Remark 1 implies that the interpolation problem (17) is a limit of a sequence of Lagrange interpolation problems.

For general (A,Γ) the interpolation conditions (16) at $\mathbf{v} \in V(A,\Gamma)$ are determined by the structure of a chosen basis of $\mathcal{K}(A_{\mathbf{v}})$. The construction of such bases is discussed by Dahmen (this volume) and by deBoor and Ron (1988).

3. Examples

The first two examples are in R^2 and can be displayed graphically. We consider two Lagrange interpolation problems, for the same set of directions A, and then two Hermite-type problems, obtained as limits of the Lagrange problems.

Example 1 Let $A = \{\mathbf{a}^1,\ldots,\mathbf{a}^6\}$ with $\mathbf{a}^1 = \mathbf{a}^4 = (1,0)$, $\mathbf{a}^2 = \mathbf{a}^5 = (0,1)$, $\mathbf{a}^3 = \mathbf{a}^6 = (1,1)$, and let $\Gamma = \left(0,0,1,\frac{1}{2}+\varepsilon,\frac{1}{2}+\varepsilon,\frac{1}{2}\right)$ for $\varepsilon > 0$. The space $\mathcal{P}(A)$ is of dimension 12 and consists of quartic polynomials which reduce to cubics along hyperplanes of the form $\mathbf{a}^i \cdot \mathbf{x} = \text{const}$. The hyperplanes $\mathbf{a}^i \cdot \mathbf{x} = \gamma_i$, $i = 1,\ldots,6$ are depicted in Figure 1, together with the twelve interpolation points. Since each interpolation point belongs to exactly two hyperplanes, (A,Γ) is simple, and the data at each point is just the function value.

For $\varepsilon = 0$ the three interpolation points $\mathbf{v}^1 = (1,0)$, $\mathbf{v}^2 = (0,1)$, $\mathbf{v}^3 = (0,0)$ remain unchanged together with the corresponding $A_{\mathbf{v}^i}$. Hence also in this problem only function values are required at $\mathbf{v}^i$, $i = 1,2,3$. Each of the other three interpolation points $\mathbf{v}^4 = \left(\frac{1}{2},0\right)$, $\mathbf{v}^5 = \left(0,\frac{1}{2}\right)$, $\mathbf{v}^6 = \left(\frac{1}{2},\frac{1}{2}\right)$ is the limit of three interpolation points in the case $\varepsilon > 0$, with $A_{\mathbf{v}^i} = \{\mathbf{a}^1,\mathbf{a}^2,\mathbf{a}^3\}$, $i = 4,5,6$. Thus $\mathcal{K}(A_{\mathbf{v}^i}) = \pi_1$, $i = 4,5,6$, and the Hermite conditions are of the form

$$(f-p)(\mathbf{v}^i) = 0 \quad , \quad \frac{\partial}{\partial x_1}(f-p)(\mathbf{v}^i) = 0 \quad , \quad \frac{\partial}{\partial x_2}(f-p)(\mathbf{v}^i) = 0 \quad , \quad i = 4,5,6\,.$$

Example 2 Let A be as in Example 1 and let $\Gamma = (0,0,1,\varepsilon,\varepsilon,1-\varepsilon)$ for $\varepsilon > 0$. The space $\mathcal{P}(A)$ is as in Example 1. The hyperplanes $\mathbf{a}^i \cdot \mathbf{x} = \gamma_i$, $i = 1,\ldots,6$ are depicted in Figure 2. These hyperplanes have twelve intersection points, each belonging to exactly two hyperplanes. Thus for $\varepsilon > 0$, (A,Γ) is simple and the interpolation is of Lagrange type.

In the limit $\varepsilon \to 0$, there are only three interpolation points: $\mathbf{v}^1 = (1,0)$, $\mathbf{v}^2 = (0,1)$, $\mathbf{v}^3 = (0,0)$, each being the limit of four interpolation points in the case $\varepsilon > 0$. The

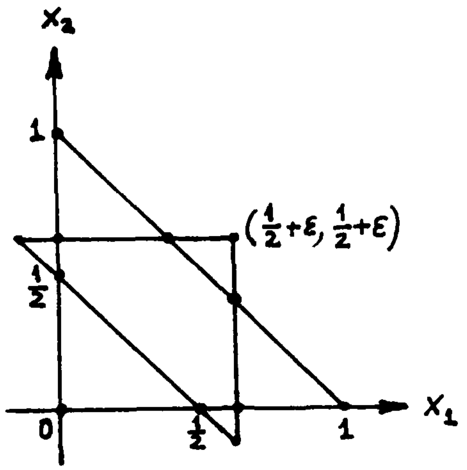


Figure 1

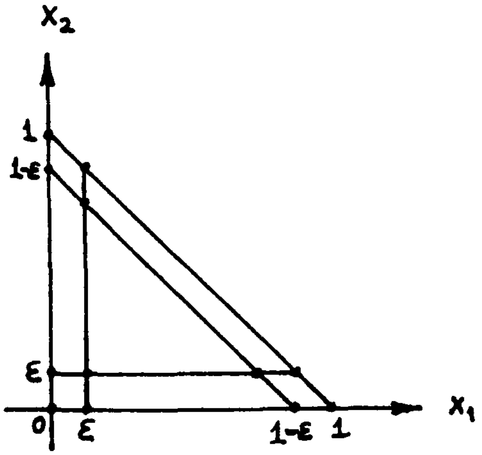


Figure 2

interpolation conditions at $\mathbf{v}^i$ are determined by $\mathcal{K}(A_{\mathbf{v}^i})$ where

$$A_{\mathbf{v}^i} = \{\mathbf{a}^j, \mathbf{a}^{j+3} \mid j \neq i\ ,\ j = 1,2,3\}\ .$$

By (15)

$$\begin{aligned}\mathcal{K}(A_{\mathbf{v}^i}) &= \{p \in \pi \mid (\mathbf{a}^j \cdot D)^2 p = 0\ ,\ j \neq i\ ,\ j = 1,2,3\} \\ &= \operatorname{span}\{1, x_1, x_2, \prod_{\substack{j=1 \\ j\neq i}}^{3} (\mathbf{n}^j \cdot \mathbf{x})\}\ ,\end{aligned}$$

where $\mathbf{n}^i \cdot \mathbf{a}^i = 0$, $i = 1,2,3$. Hence the interpolation conditions are

$$(f-p)(\mathbf{v}^i) = 0\ ,\ \frac{\partial}{\partial x_1}(f-p)(\mathbf{v}^i) = 0\ ,\ \frac{\partial}{\partial x_2}(f-p)(\mathbf{v}^i) = 0\ ,\quad i = 1,2,3$$

$$\left(\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_1\,\partial x_2}\right)(f-p)(\mathbf{v}^1) = 0 \quad , \quad \left(\frac{\partial^2}{\partial x_2^2} - \frac{\partial^2}{\partial x_1\,\partial x_2}\right)(f-p)(\mathbf{v}^2) = 0\ ,$$

$$\frac{\partial^2}{\partial x_1\,\partial x_2}(f-p)(\mathbf{v}^3) = 0\ .$$

This interpolation problem is a special case of the one solved by Gregory (1985), where the interpolation points are the vertices of a simplex in $\mathbb{R}^s$, and A consists of $s+1$ directions in general position each repeated $N \geq 2$ times. The next example deals with an extended version of this case in terms of our analysis.

Example 3 Let $B = \{\mathbf{b}^1, \ldots, \mathbf{b}^{s+1}\} \subset \mathbb{R}^s$ be in general position and let (B, Δ) be simple, with $\Delta = (\delta_1, \ldots, \delta_{s+1}) \in \mathbb{R}^{s+1}$. Given $s+1$ positive integers $m_1, \ldots, m_{s+1}$, $n = \sum_{i=1}^{s+1} m_i$, consider $A = \{\mathbf{a}^1, \ldots, \mathbf{a}^n\}$ consisting of $\mathbf{b}^i$ repeated m_i times, and $\Gamma =$

$(\gamma_1,\ldots,\gamma_n)$ consisting of δ_i repeated m_i times, $i = 1,\ldots,s+1$. The hyperplanes $H_i = \{\mathbf{x} \mid \mathbf{b}^i \cdot \mathbf{x} = \delta_i\}$, $i = 1,\ldots,s+1$, intersect at $s+1$ points $\mathbf{v}^1,\ldots,\mathbf{v}^{s+1}$, forming the vertices of a simplex. Let $\mathbf{v}^i$ denote the intersection of the s hyperplanes H_j, $j \neq i$, $j = 1,\ldots,s+1$. Then $A_{\mathbf{v}^i}$ consists of $\mathbf{b}^j$ repeated m_j times $j \neq i$, $j = 1,\ldots,s+1$, and by (15)

$$\mathcal{K}(A_{\mathbf{v}^i}) = \left\{p \in \pi \;\middle|\; (\mathbf{b}^j \cdot D)^{m_j} p \equiv 0 \;,\; j \neq i \;,\; j = 1,\ldots,s+1\right\} .$$

The dimension of $\mathcal{K}(A_{\mathbf{v}^i})$ is the number of bases in $A_{\mathbf{v}^i}$ given by $M_i = \prod_{j=1,j\neq i}^{s+1} m_j$.

Now the edge of the simplex connecting vertices $\mathbf{v}^i$ and $\mathbf{v}^\ell$ belongs to the intersection of the hyperplanes H_j, $j \neq i,\ell$, $j = 1,\ldots,s+1$. Hence $(\mathbf{v}^i - \mathbf{v}^\ell) \cdot \mathbf{b}^j = 0$, $j \neq i,\ell$, $j = 1,\ldots,s+1$, from which we conclude that any polynomial of the form

$$\prod_{\substack{\ell=1\\ \ell\neq i}}^{s+1} \left[(\mathbf{v}^i - \mathbf{v}^\ell) \cdot \mathbf{x}\right]^{\alpha_\ell} \;, \qquad 0 \leq \alpha_\ell < m_\ell \;,\; \ell = 1,\ldots,s+1 \;,$$

is annihilated by $(\mathbf{b}^j \cdot D)^{m_j}$, $j \neq i$, and therefore belongs to $\mathcal{K}(A_{\mathbf{v}^i})$. The number of these polynomials is M_i and they are linearly independent, thus forming a basis of $\mathcal{K}(A_{\mathbf{v}})$. In terms of this basis the Hermite type conditions at $\mathbf{v}^i$ are

$$\prod_{\substack{j=1\\ j\neq i}}^{s+1} \left[(\mathbf{v}^i - \mathbf{v}^j) \cdot D\right]^{\alpha_j} (f - p)(\mathbf{v}^i) = 0 \;, \quad 0 \leq \alpha_j < m_j \;,\; j \neq i \;,\; j = 1,\ldots,s+1 \;.$$

References

de Boor, C. and Ron, A. (1988), On polynomials ideals of finite codimension with applications to box spline thoery, CMS TSR # 89-21 Univ of Wisconsin-Madison.

Chui, C.K. and Lai, M.J. (1988), Vandermonde determinants and Lagrange interpolation in $\mathbb{R}^s$, in nonlinear and convex analysis, ed. by B.L. Lin, Marcel Dekker, N.Y.

Chung, K.C. and Yao, T.H. (1977), On lattices admitting unique Lagrange interpolation, SIAM J. Num. Anal. 14, 735-743.

Dahmen, W., A basis of certain spaces of multivariate polynomials and exponentials, this volume.

Dyn, N. and Ron, A. (1988), Local approximation by certain spaces of exponential polynomials, appoximation order of exponential box splines and related interpolation problems, to appear in Trans. AMS.

Gasca, M. and Maeztu, J.I. (1982), On Lagrange and Hermite interpolation in $\mathbb{R}^n$, Num. Math. **39**, 1-14.

Gregory, J.A. (1985), Interpolation to boundary data on the simplex, CAGD **2**, 43-52.

Ron, A. (1988), Exponential box splines, to appear in Const. Approx.

ALGORITHMS FOR THE CONSTRUCTION OF DATA DEPENDENT TRIANGULATIONS

NIRA DYN, DAVID LEVIN, SAMUEL RIPPA
School of Mathematical Sciences
Sackler Faculty of Exact Sciences
Tel-Aviv University

Abstract Given a set of data points in R^2 and corresponding data values it is clear that the quality of a piecewise linear interpolation over triangles depends on the specific triangulation of the data points. While conventional triangulation methods depend only on the distribution of the data points in R^2, we suggested in [1] the construction of triangulations which depend on the data values as well. In this paper we present and compare some algorithms for the construction of such data dependent triangulations.

Key words : Triangulation, Data dependent triangulation, Piecewise linear interpolation.

1 Introduction

Let $V = \{v_i = (x_i, y_i) \in R^2 , i = 1, \ldots, N\}$ be a set of distinct and non-collinear data points and $F = (F_1, \ldots, F_N)$ a (real) data vector. Suppose furthermore that $\Omega \supset V$ is a region with a polygonal boundary $\partial\Omega$ with all vertices in V.

Definition 1 *A set $T = \{T_i\}_1^t$ of non-degenerate, open, triangles is a triangulation of Ω if :*

- *V is the set of all vertices of triangles in T.*
- *Every edge of a triangle in T contains only two points from V, namely its endpoints.*
- $\overline{\Omega} = \bigcup_{i=1}^{t} \overline{T_i}$, $T_i \bigcap T_j = \emptyset$, $i \neq j$.

Given a triangulation T of Ω we consider the space $S_1^0(T)$ of piecewise linear polynomials defined over T, i.e.

$$S_1^0(T) = \{g \in C^0(\Omega) \mid \; g|_{T_i} \in \Pi_1\},$$

where Π_1 is the three dimensional space of linear polynomials. Finally we denote by f_T the unique function from $S_1^0(T)$ interpolating F, i.e.

$$f_T(x_i, y_i) = F_i \, , \; i = 1, \ldots, N.$$

A linear function is uniquely defined by its values at the three vertices of a triangle and thus a Piecewise Linear Interpolating Surface (PLIS) is uniquely determined by the choice of a specific triangulation of Ω. It is clear that the quality of approximation by a PLIS depends on the particular choice of the triangulation and naturally we look for an optimal triangulation. The classical theory says that long, thin, triangles should be avoided and that triangles should be as equiangular as possible ([4]). A popular choice of triangulation for interpolation schemes is the well known Delaunay triangulation (see e.g. [4]) which, among other nice properties, is a MaxMin triangulation, i.e. it is a triangulation T^* maximizing the quantity

$$\alpha(T) = \min_{T_i \in T}(\text{ smallest angle in } T_i).$$

The Delaunay triangulation, as various others in use, depends only on the set V and not on the data vector F. In [1] we suggested the use of *data dependent* criteria for measuring the quality of a triangulation. These criteria depend on the set V of data points and on the data vector F as well. Given a data dependent criterion and an initial triangulation, a data *dependent triangulation* T' may be constructed by the familiar procedure of swapping diagonals of convex quadrilaterals in order to get better triangulations where "better" should be interpreted as better with respect to the given data dependent criterion. Numerical tests, reported in [1], demonstrate very clearly that PLISes defined over data dependent triangulations provide better approximation, to various test functions, than the PLIS defined over the Delaunay triangulation of the same set of data points.

In the present paper we discuss algorithms for constructing data dependent triangulations. In §2 the concept of data dependent triangulations is presented as well as two algorithms for their construction. In order to simplify the presentation we use an example rather than a more detailed definition to illustrate the ideas involved. It is straightforward to extend the results to the more general setting of [1]. The most basic algorithm for the construction of data dependent triangulations is Lawson's LOP algorithm which swaps the diagonals of convex quadrilaterals in order to decrease a certain cost function. The Modified LOP (MLOP) suggested here defines the specific order of swapping edges. Two strategies for swapping edges are presented in §3 and numerical experiments comparing the LOP and MLOP algorithms are reviewed in §4.

2 Data dependent triangulations

Let V be a fixed set of data points, F a data vector, T a triangulation of Ω and suppose that f_T is the piecewise linear interpolant to F. For each interior edge e of T a real cost function $S(f_T, e)$ is assigned. and the index vector N_T, of length q, containing the cost functions of all interior edges is constructed:

$$N_T = (S(f_T, e_1), \ldots, S(f_T, e_q)).$$

The cost function of a triangulation is defined to be:

$$c(f_T) = \sum_{i=1}^{q} |S(f_T, e_i)|.$$

Two examples for cost functions are : (a) Jump in Normal Derivative (JND) $S_1(f_T, e)$, the (magnitude of the) jump in the normal derivative of f_T across the edge e and (b) Angle Between Normals (ABN) $S_2(f_T, e)$, the angle between the normal vectors to the two facets of the surface f_T on both sides of the edge e. These cost functions, and others, are discussed in more detail in [1].

Definition 2 *A triangulation T' of Ω is called optimal if $c(f_{T'}) \leq c(f_T)$ for every triangulation T of Ω.*

In most practical situations it is very difficult to obtain a globally optimal triangulation so here we consider only locally optimal triangulations. Let T be a triangulation of Ω, e an interior edge of T and Q a quadrilateral formed from the two triangles having e as a common edge. If Q is strictly convex then there are two possible ways of triangulating Q (see Figure 1).

Definition 3 *An edge e is called locally optimal if one of the following conditions holds: (a) Q is not strictly convex or (b) Q is strictly convex and $c(f_T) \leq c(f_{T'})$ where T' is obtained from T by replacing e by the other diagonal of Q.*

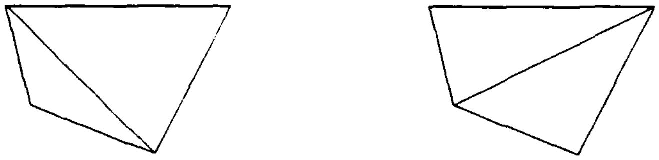


Figure 1: Two triangulations of a convex quadrilateral

Definition 4 *A locally optimal triangulation of Ω is a triangulation T' in which all edges are locally optimal.*

The basic algorithm for constructing locally optimal triangulations is the Local Optimization Procedure (LOP) suggested by Lawson [3]:

Algorithm 1 LOP

1. Construct an initial triangulation $T^{(0)}$ of Ω and set $T \leftarrow T^{(0)}$.
2. If T is locally optimal, end the procedure; else go to step 3.
3. Let e be an interior edge of T which is not locally optimal and let Q be the strictly convex quadrilateral formed from the two triangles in T having e as a common edge.
    (a) Swap diagonals of Q: Replace e by the other diagonal of Q, therefore transforming T into a triangulation T'
    (b) Set $T \leftarrow T'$ and go to step 2.

Each time an edge swap occurs, the cost function of the resulting triangulation is strictly smaller than that of the previous one. Since the number of triangulations of Ω is finite, the LOP converges, after a finite number of edge swaps, to a locally optimal triangulation.

The above LOP seems to work very well in the numerical experiments reported in [1], but it has a major drawback: the resulting locally optimal triangulation depends on the labelling of the data points and on the software implementation of the LOP. We would like to control more closely the order in which edges are swapped during the LOP iterations in order to obtain a better defined algorithm. Given a triangulation T

we consider the set $E(T)$ of all the interior edges ϵ of T which are not locally optimal. Suppose that $E(T) \neq \emptyset$ and assume that the edges are labeled such that:

$$E(T) = \{e_1, e_2, \ldots, e_{k(T)}\}.$$

A swap of some edge e_j , $1 \leq j \leq k(T)$, will transform T into a triangulation $T^{(j)}$. The Modified LOP (MLOP) selects, at each step from the set $E(T)$, the next edge to be swapped according to a predetermined swapping strategy. In §3 we discuss and compare some selection strategies. Suppose that the edge $e_p \in E(T)$ is swapped. Then $E(T^{(p)})$ has to be computed for the next iteration. This can be done efficiently since $E(T^{(p)})$ and $E(T)$ differ only in edges belonging either to the two triangles T_i and T_j which have e_p as a common edge or in edges belonging to triangles sharing an edge with T_i or T_j. Thus at most 12 edges, which may be in one of the sets and not in the other, need to be checked in each MLOP iteration (after the edge swap, e_p becomes locally optimal and thus is excluded from the set $E(T^{(p)})$).

3 Selecting the edges to swap

The first selection strategy is the *Maximal Reduction (MR)* strategy. Since a data dependent criterion selects triangulations which minimize the cost function $c(f_T)$, a natural strategy for the MLOP is to swap, in each iteration, the edge $e_p \in E(T)$ for which the $c(f_{T^{(p)}})$ is minimal, i.e.

$$c(f_{T^{(p)}}) \leq c(f_{T^{(j)}}), \;\; 1 \leq j \leq k(T).$$

The second strategy is based on the observation that often the LOP/MLOP terminates in a poor local minimum since many of the edges become interior to non-convex quadrilaterals. To avoid this we would like to swap the edges in a way which will leave the maximum possible number of convex quadrilaterals for the next MLOP iteration. Let us divide the set $E(T)$ into classes of edges according to the value of

$$m_j = I(T^{(j)}) - I(T), \;\; j = 1, \ldots, k(T),$$

where $I(T)$ is the number of convex quadrilaterals in the triangulation T. Let

$$E'(T) \subseteq E(T)$$

be the set of edges $e_j \in E(T)$ for which m_j is greatest. The *Maximal Opportunity (MO)* strategy chooses the next edge e_p to swap from the set $E'(T)$ such that $c(f_{T^{(p)}})$

is minimal, i.e.

$$c(f_{T^{(p)}}) \leq c(f_{T^{(j)}}),\ e_j \in E'(T).$$

For comparison we also generate the "worst possible" sequence of edge swaps. This is obtained by the *Minimal Reduction (MinR)* strategy which is just the opposite of the Maximal Reduction strategy, i.e. we choose the next edge to swap so that the reduction in the value of the cost function is minimized.

4 Numerical experiments

In our numerical experiments two sets of data points taken from Franke ([2]) were used. The first set contains 100 data points distributed more or less uniformly over the unit square, while the second set, with 33 data points, was designed with larger variations in the density of the data points. The data vectors $F = (F_1, \ldots, F_N)$ were obtained by evaluating various test functions, also used in [1], at the data points.

Several data dependent criteria were tested including those relating to the JND and ABN cost functions mentioned in §2 as well as others presented in [1]. For each data set and data dependent criterion, several data dependent triangulations were generated by using different strategies for swapping the edges:

- LOP - No strategy, edges are swapped according to their labelling in the edge list.
- MLOP-MR - Edges are swapped according to the Maximal Reduction strategy.
- MLOP-MO - Edges are swapped according to the Maximal Opportunity strategy.
- MLOP-MinR - Edges are swapped according to the Minimal Reduction strategy.

The Delaunay triangulation of the data points was used in all cases as an initial triangulation.

On each triangulation, the piecewise linear function f_T, interpolating the data vector F, was constructed, and the error between f_T and the test function which generates F was computed on a grid of 33×33 nodes. The mean, root mean square and maximum of these errors were tabulated along with the value of the cost function and the number of edge swaps needed for convergence of the LOP/MLOP to the data dependent triangulation.

In the numerical experiments it became clear that the order in which edges are swapped during LOP/MLOP iterations may have a large influence on the final locally optimal triangulation in terms of the quality of approximation to the test functions and the value of the cost function of the locally optimal triangulation. It was found that the value of the JND and ABN cost functions of a triangulation is usually a good indicator to the quality of approximation to the test functions: a PLIS defined over a low cost triangulation (for which these cost functions have small values) usually provides a better approximation than a PLIS defined on a high cost triangulation.

The MLOP with the MR edge swapping strategy performed quite well in general. The resulting data dependent triangulations were, in most cases, comparable or better than data dependent triangulations resulting from other strategies of edge swapping. When we say "better" we mean that a better approximation to the test functions was achieved. The MLOP-MR triangulation was in most cases the triangulation which attains the smallest value of the cost function. Also the MLOP-MR converges in the fewest number of edge swaps and thus is the most efficient. In view of these reasons this is the scheme of our choice.

The MLOP-MO strategy was comparable to the MLOP-MR strategy but is more difficult to program and thus we do not recommend the use of it.

The MLOP with the MinR strategy did produce in most cases the worst data dependent triangulation of all and used the largest number of edge swaps. The MLOP-MinR triangulation demonstrates that there is a sequence of edge swaps which can lead to very bad data dependent triangulations.

As can be expected from the above observations, the LOP swapping according to labelling resulted in good and bad triangulations depending on the labelling used, the triangulation criterion and the test function. The results are usually quite acceptable but often worse than the MLOP-MR triangulation and sometimes much worse. There are advantages, however, in using the LOP since its programming is simpler and it requires less computer storage than the MLOP for which the list $E(T)$ of edges has to be stored and maintained.

We note that none of the above strategies performed well in all cases. For any of these strategies there are examples of poor data dependent triangulations generated by it. It may be interesting to look for other heuristic strategies for edge swapping.

An example of data dependent triangulation, taken from [1], concludes this paper. The cliff function $F = (\tanh(9y - 9x) + 1)/9$ (see Figure 2), was sampled on a set of 33 data points. The Delaunay triangulation of the set of data points and the related PLIS are displayed in Figure 3. A data dependent triangulation, based on the JND cost function of §2, and the PLIS defined over it are displayed in Figure 4. These pictures demonstrate very clearly the advantage in using data dependent triangulations.

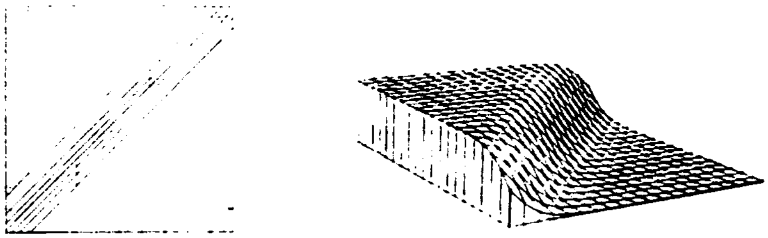


Figure 2: The cliff function

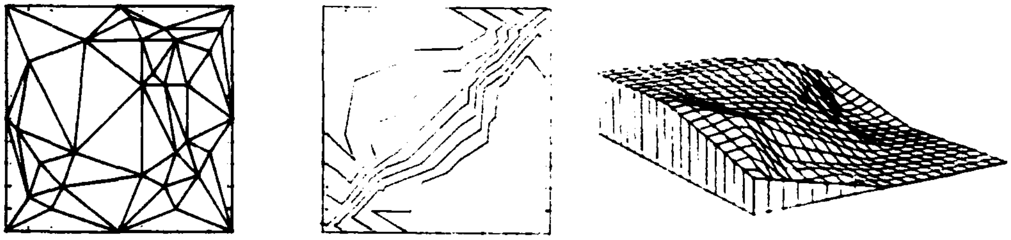


Figure 3: The Delaunay triangulation and the PLIS defined over it

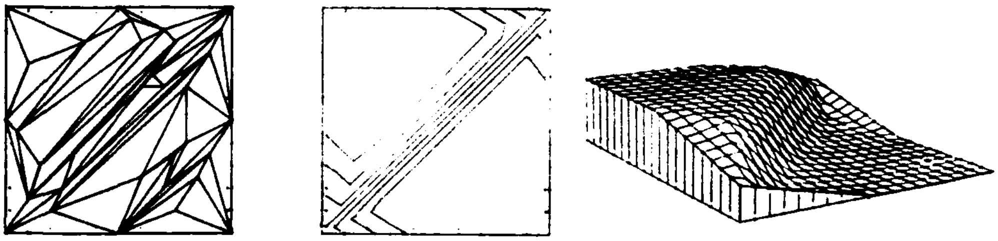


Figure 4: A data dependent triangulation and the PLIS defined over it

References

[1] Dyn, N., Levin, D. and Rippa, S. *Data dependent triangulations for piecewise linear interpolation.* To appear in IMA J. Numer. Anal.

[2] Franke, R., *Scattered data interpolation: tests of some methods,* Math Comp. **38** (1982), 181-200.

[3] Lawson, C. L., *Software for C^1 interpolation,* in Mathematical Software III, J. R. Rice (ed.), Academic Press, New York, 1977, 161-194.

[4] Schumaker, L. L., *Triangulation Methods,* in Topics in Multivariate Approximation, C. K. Chui, L. L. Schumaker and F. I. Utreras (eds.), Academic Press, 1987, 219-232.

æ

ALGORITHMS FOR COMPUTING BEST PARAMETRIC CUBIC INTERPOLATION

C. RADEMACHER - K. SCHERER
Institut für Angewandte Mathematik
Universität Bonn, W-Germany

Abstract We consider the problem of finding a curve passing through prescribed points in $\mathbb{R}^d$ such that the kinetic energy of particle moving along this curve is minimized. Recently uniqueness of the solution of this problem has been established by the second author. In this contribution algorithms for computing the solution are devised and their (global) convergence properties studied.

Key words: Best interpolation, cubic spline curves, constrained minimization, stationary points, projected gradient method, Newton Method.

1. Introduction. The problem of best parametric interpolation of given data $\{\underline{y}_i\}_{i=1}^n$ in $\mathbb{R}^d$ is the following: find a function $\underline{s}(t) \in \mathcal{L}_2^k(0,1)$, the space of $\mathbb{R}^d$-valued functions on [0,1] with components in $L_2^k(a,b)$, which attains

$$\inf_{\underline{t}} \inf \{\int_0^1 ||\underline{f}^{(k)}(t)||^2 dt \ : \ f \in \mathcal{L}_2^k(0,1), \underline{f}(t_i)=\underline{y}_i \ , \ 1 \le i \le n\}. \quad (1)$$

The first infimum runs over all sequences $\underline{t}$ of nodes satisfying

$$0 = t_1 < t_2 < \ldots < t_n = 1, \quad (2)$$

and the norm $\| \ \|$ in (1) denotes the usual Euclidean norm in $\mathbb{R}^d$. If one considers only tne second infimum in (1) the problem reduces to the classical problem of best interpolation in spline theory. For the motivation of the general problem we refer to [5], [3] and [4]. In [3] and [4] the uniqueness of the solution of (1) in the cubic case $k=2$ is proved ($d=1$ in [3] and general d in [4]) under the condition

$$\underline{y}_i \neq \underline{y}_{i+1}, \qquad 1 \le i \le n-1, \tag{3}$$

on the data.

In this note we describe two types of convergent algorithm for computing the solution $\underline{s}^*(t)$ of (1) in the case $k=2$. First we summarize some well-known facts and the relevant results of [4]. For fixed $\underline{t}$ the inner infimum in (1) is attained by a cubic spline function $\underline{s}(t)$ whose second derivative is

$$\underline{s}''(t) = \sum_{i=1}^{n-2} \underline{a}_i \, N_{i,2}(t), \tag{4}$$

where the $N_{i,2}(t)$ denote the piecewise linear B-splines with knots in $\underline{t}$ satisfying

$$N_{i,2}(t_j) = \delta_{j,i+1}, \qquad 1 \le j,i \le n-2 \ .$$

The coefficients $\underline{a}_i \in \mathbb{R}^d$ in (4) are therefore equal to $\underline{s}''(t_{i+1})$. They are determined by the tridiagonal lincar system

$$\sum_{i=1}^{n-2} G_{ij} \, \underline{a}_i = \underline{z}_j := \frac{\underline{w}_{j+1}}{h_{j+1}} - \frac{\underline{w}_j}{h_j}, \qquad \underline{w}_j := \underline{y}_{j+1} - \underline{y}_j \quad , \tag{5}$$

where

$$G_{ij} = \begin{cases} 2(h_i + h_{i+1}), & j = i, \\ h_{\max(i,j)} \quad , & |i-j| = 1 \ , \ h_i := t_{i+1} - t_i, \\ 0 \quad , & \text{otherwise}. \end{cases} \tag{6}$$

We can write (5) in matrix form as

$$G\,\underline{A} = \underline{Z} \tag{7}$$

if we introduce the vectors

$$\underline{A} := (\underline{a}_1,\ldots,\underline{a}_{n-2}), \qquad \underline{Z} = (\underline{z}_1,\ldots,\underline{z}_{n-2}).$$

Furthermore one can easily verify

$$\int_0^1 \|\underline{s}''(t)\|^2 dt = <\underline{A}\,,G\underline{A}> \;=<\underline{A}\,,\underline{Z}>, \tag{8}$$

where $<\ ,>$ denotes the obvious scalar product. From this one concludes that the infimum problem (1) is equivalent to

$$\inf_{\underline{h}\in K}\ \{F(\underline{h})\ :\ F(\underline{h}) := <\underline{A}\,,\underline{Z}>,\quad G\underline{A} = \underline{Z}>, \tag{9}$$

where

$$K := \{\underline{h}\in\mathbb{R}^{n-1}\ :\ \underline{h} = (h_1,\ldots,h_{n-1})\,,\,h_i > 0,\quad \sum_{i=1}^{n-1} h_i = 1\}. \tag{10}$$

Hence a solution of (1) is given by a pair $(\underline{h}\,,\underline{A}(\underline{h}))$, where $\underline{A} = A(\underline{h})$ satisfies (7) and $\underline{h}$ the equations for a critical point of $F(\underline{h})$ in K. These equations read

$$0 = \lambda + \frac{\partial F}{\partial h_j} = \lambda - \frac{2T_j}{h_j^2} - \alpha_j, \qquad 1\le j\le n-1, \tag{11a}$$

$$1 = \sum_{j=1}^{n-1} h_j, \tag{11b}$$

with Lagrangian parameter $\lambda\in\mathbb{R}$. For the quantities in (11) one has the explicit expressions (with $\underline{a}_o = \underline{a}_{n-1} = 0$)

$$T_j := -\,h_j^2 < \underline{A}\,,\frac{\partial Z}{\partial h_j} > = (\underline{a}_{j-1} - \underline{a}_j\,,\underline{w}_j), \tag{12a}$$

$$\alpha_j := < \underline{A}, \frac{\partial G}{\partial h_j} \underline{A} > = \frac{1}{3} [(\underline{a}_j,\underline{a}_j) + (\underline{a}_{j-1},\underline{a}_{j-1}) + (\underline{a}_j,\underline{a}_{j-1})]. \quad (12b)$$

Here (,) denotes the usual scalar product in $\mathbb{R}^d$.

The equations (11) together with (7) form a nonlinear system of equations for $\underline{h} \in K$. In [4] the following was proved:

<u>Theorem 1:</u> There exists only one minimum $\underline{h}^*$ of F(h) in K (which is then the unique, global, solution of (1)). The corresponding pair $\underline{h}^*$, $\underline{A}^* = A(\underline{h}^*)$ is a solution of the saddle point problem

$$\inf_{\underline{h} \in K} \sup_{\underline{A}} \phi (\underline{A},\underline{h}) = \sup_{\underline{A}} \inf_{\underline{h} \in K} \phi (\underline{A},\underline{h}), \quad (13)$$

where $\phi (\underline{A},\underline{h}) := - < \underline{A}, G\underline{A} > + 2 < \underline{A},\underline{Z} >$. A critical point of $F(\underline{h})$, i.e. a solution of (11) and (7), is a solution of (1) if and only if the corresponding A* lies in

$$B := \{\underline{A} \in \mathbb{R}^{(n-2)d} : T_j := (\underline{a}_{j-1} - \underline{a}_j, \underline{w}_j) > 0, \ 1 \leq j \leq n-1\}. \quad (14)$$

2. Descent Methods. In the following we describe how some descent methods for constrained minimization problems may be adapted and modified to problem (9) so as to guarantee global convergence. A general method for minimization with linear equality constraints has the form:

Given the approximation $\underline{h}^{(\nu)}$ for the solution $\underline{h}^* \in \mathbb{R}^{n-1}$, one computes an (improved) approximation $\underline{h}^{(\nu-1)}$ for $\underline{h}^*$ by

$$\underline{h}^{(\nu+1)} := \underline{h}^{(\nu)} + \delta_\nu P \underline{d}^{(\nu)}, \qquad \nu = 0,1,\ldots, \quad (15)$$

where $\underline{d}^{(\nu)}$ is the direction of search $(||\underline{d}^\nu|| = 1)$, the positive scalar δ_ν the step length, and P the linear projection of $\underline{d}^{(\nu)}$ to ker R, where the constraints are given in the form $R\underline{h} = \underline{b}$ with matrix R. In the case of problem (9) a simple choice is to set

$$(P\underline{d})_j := d_j - \frac{1}{(n-1)} \sum_{i=1}^{n-1} d_i \quad (16)$$

for the vector $\underline{d} = (d_1,\dots,d_{n-1})$.

In order that the above method is a descent method we assume that there is a number $\beta > 0$, independent of $\nu = 0,1,2,\dots,$ such that (note that $P = P^T$ with the choice (16))

$$\left(\underline{g}^{(\nu)}, P\underline{d}^{(\nu)}\right) \leq - \beta ||P^T g^{(\nu)}||, \tag{17}$$

where $g^{(\nu)}$ denotes the gradient of F at $\underline{h}^{(\nu)}$, i.e.

$$\underline{g}^{(\nu)} := \text{grad } F\left(\underline{h}^{(\nu)}\right).$$

(Note that here (,) stands for the scalar product in $\mathbb{R}^{n-1}$.) Then Taylor expansion gives the decrease of $F(\underline{h}^{(\nu+1)})$ for δ_ν sufficiently small. The following step-length procedure realizes this.

Choose $\delta_\nu = 2^{-j}$, where $j = j(\nu)$ is the smallest non-negative number such that

$$F\left(\underline{h}^{(\nu)} + 2^{-j}\underline{d}^{(\nu)}\right) < F\left(\underline{h}^{(\nu)}\right) + 2^{-j-1}\left(\underline{g}^{(\nu)}, P\underline{d}^{(\nu)}\right) \tag{18a}$$

is valid. In this procedure the constraints $0 < h_i^{(\nu+1)} < 1$, $1 \leq i \leq n-1$, implied by $\underline{h}^{(\nu+1)} \in K$ can also be incorporated. To this end we require additionally that $\delta_\nu = 2^{-j}$ shall satisfy

$$-h_i^{(\nu)} < 2^{-j}\left(P\underline{d}^{(\nu)}\right)_i < 1 - h_i^{(\nu)}, \qquad 1 \leq i \leq n-1 . \tag{18b}$$

Both (18a) and (18b) are always satisfied for $\delta_\nu = 2^{-j}$ sufficiently small. This is guaranteed by the fact that for any $\underline{h} \in K$ the inequality

$$\min_j h_j \geq \min_i ||\underline{y}_{i+1} - \underline{y}_i||^2 / [(n-1)\max_j ||\underline{y}_{j+1} - y_j||^2 + F(\underline{h})] \tag{19}$$

holds (see [4]).

The above considerations only ensure convergence to a critical point. In order to improve this we introduce so called "T steps" for $\underline{h}^{(\nu)} \in K$ such that not all corresponding T_i's defind by (12a) are positive. In this case a critical point $\underline{h}^*$ of F(h) cannot be the solution of (1) according to (14). Suppose it is not a strict local minimum too. Then

$$\eta = \min_{||\underline{d}||=1} \left(P\underline{d}\ , H(\underline{h}^*)P\underline{d}\right) \leq 0 \qquad (20)$$

must hold,where $H(\underline{h})$ denotes the Hessian of $F(\underline{h})$ at a point $\underline{h}$. If $\underline{d}^*$ is a direction attaining η in (20),the decrease of $F(\underline{h})$ along $\underline{d}^*$ satisfies, for $0 < \delta \leq \delta_o$,

$$F(\underline{h}^* + \delta P\underline{d}^*) \leq F(\underline{h}^*) + \max\,(-C\delta^{k_o}, \eta\,\delta^2) \qquad (21)$$

with some constant $C > 0$ and a natural number k_o. This can be seen from Taylor expansion about $\underline{h}^*$.

Therefore we define a "T-step" as follows: for an estimate $\underline{h}^{(\nu)}$ compute a direction $\tilde{d}^{(\nu)}$ for which

$$\eta^{(\nu)} = \left(P\underline{\tilde{d}}^{(\nu)}\ , H(\underline{h}^{(\nu)})\ P\underline{\tilde{d}}^{(\nu)}\right) = \min_{||\underline{d}||=1} \left(P\underline{d}\ , H(\underline{h}^{(\nu)})P\underline{d}\right). \qquad (22)$$

The step length $\tilde{\delta}_\nu$ for this direction is chosen as $\tilde{\delta}_\nu = 2^{-\ell_o}$ and ℓ_o by

$$\varphi(2^{-\ell_o}) = \min\{\varphi(2^{-\ell})\ :\ \varphi(2^{-\ell}) \leq F(\underline{h}^{(\nu)} + \delta_\nu Pd^{(\nu)})\}. \qquad (18c)$$

Here each j has to satisfy (18b), and $\varphi(t)$ is defined by $\varphi(t) := F(\underline{h}^{(\nu)} + t\,P\tilde{d}^{(\nu)})$. We remark that (18c) can be satisfied only if some T_i are non-positive. Therefore we combine the "T-steps" with the "normal steps" to obtain the following algorithm:

Given an estimate $\underline{h}^{(\nu)}$ for the solution $\underline{h}$, compute

(i) the vector $A^{(\nu)} = \underline{A}(\underline{h}^{(\nu)})$ from (7) with $G = G(\underline{h}^{(\nu)})$, $Z = Z(\underline{h}^{(\nu)})$,

ii) a direction of search $\underline{d}^{(\nu)}$ and its projection $P\underline{d}^{(\nu)}$ according to (16) such that (17) is satisfied, and a step length σ_ν according to (18a), (18b),

(iii) the new estimate $\underline{h}^{(\nu+1)}$ via formula (15), if all T_i formed in (12a) with respect to $\underline{A}^{(\nu)}$ are positive, otherwise

iv) a direction of search $\underset{\sim}{\tilde{d}}^{(\nu)}$ according to (22), and the test (18c),

v) the new estimate $\underline{h}^{(\nu+1)}$ by setting $\underline{h}^{(\nu+1)} := \underline{h}^{(\nu)} + \tilde{\sigma}_{\nu}\underline{\tilde{d}}^{(\nu)}$

if the test is positive, otherwise define $\underline{h}^{(\nu+1)}$ as in iii).

We can now prove

<u>Theorem 2:</u> The above algorithm converges under the assumotion (18) for any starting vector $\underline{h}^{(o)} \in K$ to a strict local minimum of problem (9).

<u>Proof</u> By (19) all the iterates $\underline{h}^{(\nu)}$ lie in a compact subset of K. Hence there is a subsequence converging to some $\underline{h}^* \in [0,1]^n$ which we denote by $\{\underline{h}^{(j\nu)}\}_{\nu=1}^{\infty}$.

Following standard lines (cf. [2])it can be established that $\underline{h}^*$ is a critical point of $F(\underline{h})$. But we have still to show that $\underline{h}^*$ is a strict local minimum of (9). To this end we choose for $\varepsilon > 0$ suitable $\nu_o = \nu_o(\varepsilon)$ so large that for all $\varepsilon \geq \nu_o$

$$||\underline{h}^{(j\nu)} - \underline{h}^*|| \leq \varepsilon, \qquad |||H(\underline{h}^{(j\nu)}) - H(\underline{h}^*)||| \leq \gamma_1(\varepsilon), \quad \nu \geq \nu_o,$$

where $|||\cdot|||$ denotes the matrix norm on $\mathbb{R}^{n-1}$ and $\gamma_1(\varepsilon)$ is a constant tending to zero for $\varepsilon \to 0$. Then we can find $\gamma_2(\varepsilon)$ with $\gamma_2(\varepsilon) \to 0$ for $\varepsilon \to 0$ such that, for $\nu \geq \nu_o$,

$$F\left(h^{(j\nu)} + \delta P \underline{\tilde{d}}^{(j\nu)}\right) \leq F\left(\underline{h}^* + \delta P \underline{d}^*\right) + \gamma_2(\varepsilon), \qquad (23)$$

where $\underline{d}^*$ is defined via (20). This is a consequence of the fact that η and $\underline{d}^*$ depend continously on $\underline{h}^*$.

We assume then that $\underline{h}^*$ is not a strict local minimum of (9), i.e. (20) holds. Then it follows from (23) and (21) (for $\tilde{\sigma}_{j\nu} \leq \delta_o$) that

$$\varphi(2^{-\ell}) = F\left(\underline{h}^{(j\nu)} + 2^{-\ell}\underline{\tilde{d}}^{(j\nu)}\right) \leq F(\underline{h}^*) + \max\left(-C2^{-\ell k_o}, \eta 4^{-\ell}\right) + \gamma_2(\varepsilon)$$

$$\le F\left(\underline{h}^{(j\nu)} + \sigma_{j\nu} P\underline{d}^{(j\nu)}\right) + \max\left(-C2^{-\ell k_o}, \eta 4^{-\ell}\right) + \gamma_2(\varepsilon).$$

This means that in (18c) a finite step length $\tilde{\sigma}_{j\nu} = 2^{-\ell_o} \le \delta_o/2$ will be chosen provided ε – and hence $\nu_o = \nu_o(\varepsilon)$ – are chosen such that $\gamma_2(\varepsilon) \le (1/2)\min(C,-\eta)$. Then we obtain a contradiction since ℓ_o is such that

$$F\left(\underline{h}^{(j\nu)+1}\right) = \varphi(2^{-\ell_o}) \le F(\underline{h}^*) + (1/2)\max(-c\ ,\ \eta).$$

As an application of this theorem we state

Corollary: The above algorithm converges to a strict local minimum if one takes as direction of search $\underline{d}^{(\nu)}$ the direction of the projected gradient $Pg^{(\nu)}$.

3. Further methods. The proposed modification of the projected gradient method is not yet practical, since a"T step" requires too much work. This is due to the fact that the Hessian of $F(\underline{h})$ is given only implicitly , namely by (cf.[4])

$$\frac{\partial^2 F}{\partial h_i \partial h_j} = < \frac{\partial A}{\partial h_i}\ ,\ G\,\frac{\partial A}{\partial h_j} > + \delta_{ij}\, T_i h_i^{-3}\ ,$$

where the vectors $\{\partial A/\partial h_i\}_{i=1}^{n-1}$ have to be determined by the systems

$$G(\partial \underline{A}/\partial h_i) = (\partial \underline{Z}/\partial h_i) - (\partial \underline{G}/\partial h_i)\,\underline{A}. \qquad (24)$$

Thus the amount of work of a "T step" is n-times larger than that of a usual one if one assumes that the essential work is solving the $(n-2) \times (n-2)$ tridiagonal systems (with always the same matrix G). Namely,a "normal"step requires the solution of d such systems (see (5), (9) for the evaluation F(h))whereas a "T step" needs (n-1)d additional solutions.

In the actual computation we used a simplified version of a "T step" based on the use of directions $\underline{e}^{(\ell)}$ defined via

$$e_i^{(\ell)} := \begin{cases} -1, & i = \ell, \\ 1, & i = -\ell, \\ 0 & \text{otherwise}, \end{cases}$$

and the following definition of $\tilde{d}^{(\nu)}$ in the above algorithm:

$$\underline{\tilde{d}}^{(\nu)} := \sum_{\ell \in I^{(\nu)}} \underline{e}^{(\ell)}, \quad I^{(\nu)} := \{\ell \in \{2,\dots,n-1\} : T_\ell^{(\nu)} \leq 0, \; T_{\ell-1}^{(\nu)} > 0\}, \qquad (25)$$

where the $T_\ell^{(\nu)}$ are defined via (12a) with respect to $\underline{A}^{(\nu)} \equiv \underline{A}(\underline{h}^{(\nu)})$. The reason for the restriction $\ell \in I^{(\nu)}$ is that in the first order expansion $(h_\ell \equiv h_\ell^{(\nu)})$,

$$F\left(\underline{h}^{(\nu)} + \delta \underline{\tilde{d}}^{(\nu)}\right) \approx F\left(\underline{h}^{(\nu)}\right) + \delta \sum_{\ell \in I^{(\nu)}} \left[\alpha_\ell^{(\nu)} - \alpha_{\ell-1}^{(\nu)} + 2T_\ell^{(\nu)} h_\ell^{-2} - 2T_{\ell-1}^{(\nu)} h_{\ell-1}^{-2}\right],$$

the negative part with the $T_\ell^{(\nu)}$, $T_{\ell-1}^{(\nu)}$ should dominate.

We used direction (25) combined with the step length procedure (18b), (18c) in the above algorithm successfully despite the fact that we could not prove that it is a direction of descent near a critical point other than the solution of (1). Our experience was that it usually produced a larger decrease of F(h) than the (projected) direction of steepest descent. The effectivness of this technique is highlighted by the fact that only with its help the solution of (1) could be found in presence of another critical point (see example 2 below). Here one should remark that this statement is not quite precise in that even Theorem 2 only guarantees convergence to a strict local minimum of $F(\underline{h})$. However it is very likely (with homotypy arguments) that there is only one such minimum, namely the global one.

Another idea is to use Newton's method. We omit the details since each iteration would require the solution of (n-1)d tridiagonal systems for the computation of the Hessian (cf. (24)). Instead we used a less expensive variant based on a different parametrization of $F(\underline{h})$ which will be described shortly. Introducing

$$\underline{s}_i := s'(t_i), \quad \underline{S} := (\underline{s}_1, \ldots, \underline{s}_n),$$

we obtain after some computation, instead of (8), the representation

$$\int_0^1 \| \underline{s}''(t) \|^2 dt = 12 \left[\sum_{i=1}^{n} \| \underline{w}_i \|^2 h_i^{-3} - < \underline{S}, \underline{R} > + < \underline{S}, \Gamma \underline{S} > \right] = \Psi(\underline{h}, \underline{S}), \tag{26}$$

where $\underline{R} := (\underline{r}_1, \ldots, \underline{r}_n)$ is given by

$$\underline{r}_j := \begin{cases} \underline{w}_1 \, h_1^{-2} & , \quad j = 1 \\ \underline{w}_j \, h_j^{-2} + \underline{w}_{j-1} \, h_{j-1}^{-2}, & 2 \le j \le n-1 \\ \underline{w}_{n-1} \, h_{n-1}^{-2} & , \quad j = n \end{cases}$$

and $\Gamma = (\Gamma_{ij})$ denotes the symmetric $n \times n$ matrix

$$\Gamma := \frac{1}{3} \begin{pmatrix} 2h_1^{-1} & h_1^{-1} & & & \\ h_1^{-1} & \ddots & \ddots & & \\ & h_{i-1}^{-1} & 2(h_i^{-1} + h_{i-1}^{-1}) & h_i^{-1} & \\ & & \ddots & \ddots & \ddots \\ & & & h_{n-1}^{-1} & 2h_{n-1}^{-1} \end{pmatrix}. \tag{27}$$

Since any cubic spline function with $\underline{s}(t_i)$, $\underline{s}'(t_i)$ prescribed for $1 \le i \le n$ is feasible in the infimum of (1), this problem is equivalent to

$$\inf_{\underline{h} \in K, \underline{S} \in \mathbb{R}^{nd}} \Psi(\underline{h}, \underline{S}).$$

The equations for a critical point $\underline{h} \in K, \underline{S}$ of $\Psi(\underline{h}, \underline{S})$ read

$$0 = \mu + \frac{\partial \Psi}{\partial h_j} = \mu - \frac{36}{h_j^4} \| \underline{w}_j \|^2 + \frac{24}{h_j^3} (\underline{w}_j, \underline{s}_{j+1} + \underline{s}_j) - \\ - \frac{4}{h_j^2} \left[\| \underline{s}_j \|^2 + \| \underline{s}_{j+1} \|^2 + (\underline{s}_j, \underline{s}_{j+1}) \right], \tag{28}$$

$$1 = \sum_{j=1}^{n-1} h_j, \tag{28b}$$

$$0 = \frac{\partial\Psi}{\partial\underline{S}} = -\,12\underline{R} + 24\,\Gamma\,\underline{S}, \tag{28c}$$

with Lagrangian parameter μ. The notation $\partial\Psi/\partial\underline{S}$ stands for the gradient with respect to the variables in $\underline{S}$. It is well known ([1, chapter 4]) that (28c) is a defining system for the natural cubic interpolating spline curve in $\mathbb{R}^d$. Hence the solutions of equations (8), (11) in $\underline{h}$, $\underline{A}$ are same as those in (28).

In order to compute the Jacobian of (28) the $n \times d$ unknowns of these $n \times d$ equations are ordered as

$$h_1,\dots,h_{n-1},h_n := \mu\,;\ \underline{S} = (\underline{S}_1,\dots,\underline{S}_d)\ ;\ \underline{S}_\ell := (s_i^{(\ell)},\dots,s_n^{(\ell)}) \in \mathbb{R}^n.$$

The associated $n \times d$ mappings are ($\underline{h}$ denotes now the old $\underline{h}$ enlarged by h_n)

$$\begin{aligned}
\varphi_i(\underline{h},\underline{S}) &:= h_n + \partial\Psi/\partial h_i, \quad 1 \le i \le n-1,\\
\varphi_n(\underline{h},\underline{S}) &:= \sum_{i=1}^{n-1} h_i - 1, \quad i = n, \qquad (29)\\
\phi_{i,\ell}(\underline{h},\underline{S}) &:= \partial\Psi/\partial s_i^{(\ell)}, \qquad 1 \le i \le n,\ 1 \le \ell \le d.
\end{aligned}$$

The Jacobian J of this mapping on $\mathbb{R}^{nd}$ into $\mathbb{R}^{nd}$ then has the block form

$$J = \left(\begin{array}{c:c:c:c}
E & B_1 & & B_d \\ \hdashline
B_1^T & \Gamma & & \\ \hdashline
 & & & \\ \hdashline
B_d^T & & & \Gamma
\end{array}\right), \qquad E = \begin{pmatrix}
\eta_1 & & & 1 \\
 & \ddots & & \vdots \\
 & & \eta_{n-1} & 1 \\
1 & \cdots & 1 & 0
\end{pmatrix},$$

where the numbers η_j are given by

$$\eta_j = 144\|\underline{w}_j\|^2 h_j^{-5} - 72(\underline{w}_j, \underline{s}_j + \underline{s}_{j+1}) h_j^{-4} + 8h_j^{-3}\Big[\|\underline{s}_j\|^2 + \|\underline{s}_{j+1}\| + (\underline{s}_j, \underline{s}_{j+1})\Big]. \tag{30}$$

The only nonzero entries of the matrices B_ℓ are

$$b_{ii}^{(\ell)} = 24 w_i^{(\ell)} h_i^{-3} - 8 s_i^{(\ell)} h_i^{-2} - 4 s_{i+1}^{(\ell)} h_i^{-2}, \qquad 1 \le i \le n-1,$$

$$b_{i,i+1}^{(\ell)} = 24 w_i^{(\ell)} h_i^{-3} - 8 s_{i+1}^{(\ell)} h_i^{-2} - 4 s_i^{(\ell)} h_i^{-2}, \qquad 1 \le i \le n-1,$$

and the matrix Γ is defined by (27). With the knowledge of J the Newton method for solving the system (30) can be set up in the standard manner. We omit the details due to the lack of space. We only mention that by using the block structure of J the solution of the linear equations with matrix J can be reduced to the solution of d systems with matrix Γ and one system with matrix $E - \sum_{\nu=1}^{d} B_\nu \Gamma^{-1} B_\nu^T$.

In the actual computation we simplified J by setting the matrices $B_1^T, \ldots, B_d^T$ equal to zero. Then the solution of the resulting system requires the solution of d systems with Γ and one system with E. Thus the amount of work for computing a new direction of search is comparable to that of the projected gradient method above. We then formed a descent method with this direction of search and step length procedure as above together with "T steps" in the version of (25). We could not prove global convergence of this Quasi-Newton method. But we have at least local convergence in view of the relation

$$\eta_j = 12 T_j h_j^{-3} + 8h_j^{-3}\Big[\|\underline{s}_j\|^2 + \|\underline{s}_{j+1}\|^2 + (\underline{s}_j, \underline{s}_{j+1})\Big],$$

which follows from (30) and (12a). Specifically in view of (14), all these numbers are strictly positive near the solution and hence E has a bounded inverse.

4. Numerical results. Many examples were computed in order to compare the behaviour of the projected gradient method (PG) with that of the Quasi-Newton method (QN). As starting points for the algorithms we chose either the uniform parametrization (U) with knots $h_i = 1/(n-1)$ or the normalized accumulated chord length parametrization (CL).

It was observed that for the PG method a start with CL parametrization often took fewer iterations for convergence, whereas for the QN method this happened for the U parametrization. Concerning the dependence of the data it turned out that for "smooth" data (cf. the data in S-form in [3]) the QN method often needed more iterations than the PG method but even fewer function evaluations (which correspond to the number of solutions of tridiagonal systems). For data with isolated corners or peaks the QN method seems to be generally superior. The following table illustrates this (# It denote the number of iterations, # F the number of function evaluations, the tolerance for the error was chosen as 10^{-4}):

	CL		U	
QN method	# It	41	# It	114
	# F	130	# F	674
PG method	# It	332	# It	555
	# F	2737	# F	4481

Example 2

The following two examples (figures 1, 2 and 3) are chosen to illustrate the qualitative behaviour of the different parametric spline interpolants. The curved marked by ... uses the uniform, - .. - the CL and - the "optimal" parametrization. The small circles mark the data points (in figures 1 and 2 the four lower corner points are nearly identical).

The first example is particularly interesting since it exhibits the existence of a critical (or stationary) point other than the global minimum. The PG method without "T steps" terminated at a point with T_i's alternating in sign (figure 2) and at a point with

FIGURE 1 : THE MINIMUM

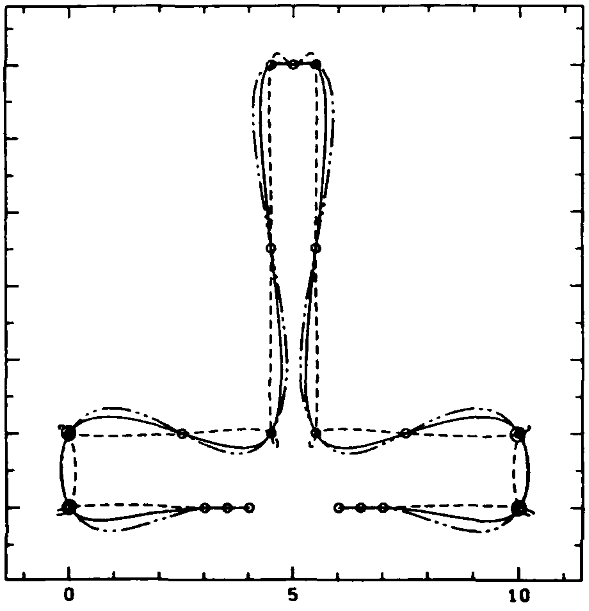


FIGURE 2 : A STATIONARY POINT

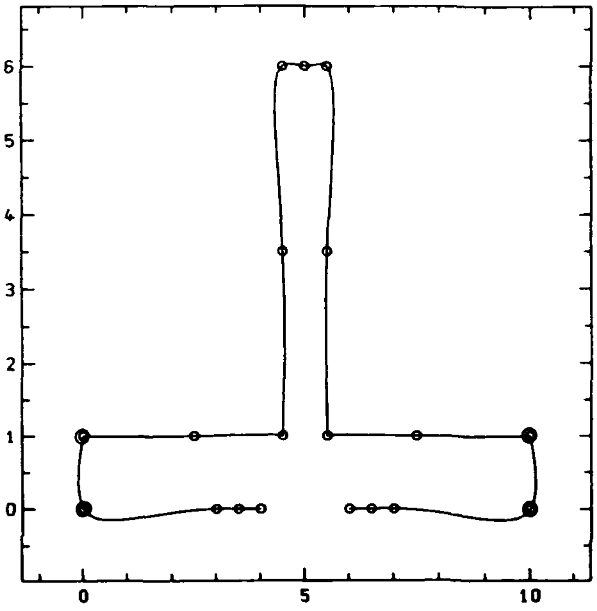


FIGURE 3

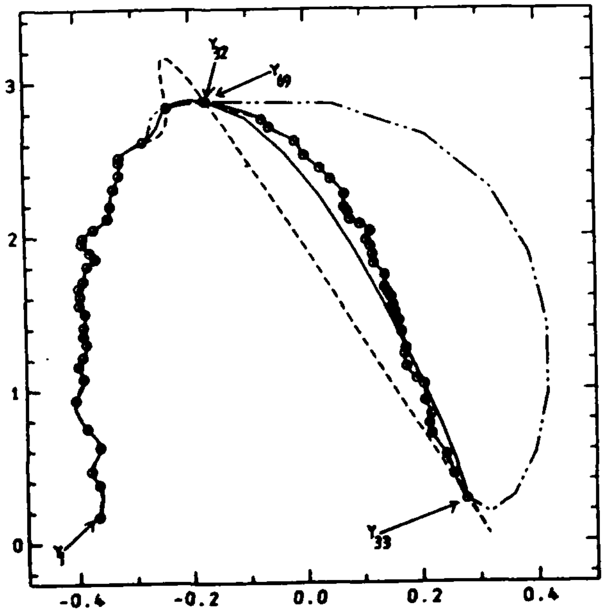


Table 1

Data of Figure 1 and Figure 2

i	y_i
1	(-1.0 , 0.0)
2	(-1.5 , 0.0)
3	(-2.0 , 0.0)
4	(-4.999 , 0.0)
5	(-5.0 , 0.001)
6	(-5.0 , 0.999)
7	(-4.999 , 1.0)
8	(-0.501 , 1.0)
9	(-0.5 , 1.001)
10	(-0.5 , 3.5)
11	(-0.5 , 6.0)
12	(0.0 , 6.0)
⋮	⋮

positive T_i if "T steps" were used (figure 1). According to (14) only the latter point is the global minimum.

The second example (figure 3) illustrates very clearly the influence of the parametrization on the resulting spline interpolant. The data points $y_1,\dots,y_{32}$ and $y_{33},\dots,y_{69}$ follow closely each other so that the curve is well described by them. Consequently the parametrization has nearly no effect; all three curves are essentially identical. However, between y_{32} and y_{33} there is a large gap in the data which provide therefore only poor information. Here the use of sound parametrization turns out to be essential.

References

[1] De Boor, C. (1978),'A practical guide to splines', Springer, New York.

[2] Gill, P.E., Murray, W., Wright, M.H. (1981), 'Practical Optimization', Academic Press, London-New York.

[3] Martin, S. (1984), 'An approach to data parametrization in parametric cubic spline-interpolation problems', J. Approx. Theory 41 (1984), 64-86.

[4] Scherer, K. (1988), 'Uniqueness of best parametric interpolation by cubic spline curves', Preprint, University of Bonn.

[5] Toepfer, H.J. (1981), 'Models for curve fitting', in: Numer. Methoden der Approximationstheorie 6 (L. Collatz ed.) ISNM 59, Birkhäuser, Basel.

4. Smoothing & Constraint Methods

DATA FITTING BY PENALIZED LEAST SQUARES

MANFRED VON GOLITSCHEK

Institut für Angewandte Mathematik

Universität Würzburg

LARRY L. SCHUMAKER

Department of Mathematics

Vanderbilt University

Abstract: In this paper we deal with the problem of approximating an unknown smooth function, given a set of data consisting of measurements on the function. We are especially interested in the case where the number of data is fairly large, and is subject to considerable noise. We focus on a method for solving this problem, called the method of *penalized least squares*, which, for some reason, does not seem to have received much attention in the Approximation Theory literature, despite the fact that it often seems to work much better than several other more commonly used methods. Our aim is to give a complete and self-contained treatment of penalized least squares, including a discussion of how to choose the smoothing parameter.

Key words: Data fitting, smoothing, least squares, penalized least squares

1. Introduction

Suppose f is an unknown function of one or more variables, and that we are given a set of measurements

$$z_i = \lambda_i f + \epsilon_i, \quad i = 1, \dots, n, \tag{1}$$

where $\lambda_1, \dots, \lambda_n$ are linear functionals defined on some linear space $\mathcal{F}$ containing f, and where $\epsilon_1, \dots, \epsilon_n$ are measurement errors. Typically the λ_i will be point-evaluation of the function or one of its derivatives, although other functionals are also of interest.

Our aim is to approximate f using a finite dimensional subspace $\mathcal{S}$ of $\mathcal{F}$ spanned by functions $B_1, \dots, B_k$, where $k \le n$. For each $c := (c_1, \dots, c_k)^T$, let

$$s_c = \sum_{i=1}^{k} c_i B_i. \tag{2}$$

The problem is to devise some reasonable means of choosing the coefficients. Perhaps the best-known approach is the classical method of *least-squares* whereby we minimize

$$E_z(c) = \frac{1}{n}\sum_{i=1}^{n}[\lambda_i s_c - z_i]^2 \tag{3}$$

over all choices of c.

While straightforward least-squares fitting is often appropriate, for some data fitting problems it produces a function which is not sufficiently smooth (for an example, see [11]). In such cases, it may be better to look for a function in $\mathcal{S}$ which minimizes a combination of the goodness of fit (3) with an appropriate measure of smoothness of the fitting function.

Many reasonable measures of smoothness of the function s_c can be expressed in the form

$$J(c) = c^T E c, \tag{4}$$

where E is a symmetric nonnegative-definite $k \times k$ matrix. For typical examples in univariate and bivariate data fitting, see Section 5 and Remark 5, respectively.

Suppose now that $\lambda \geq 0$ is given and that $E \neq 0$. Then for each $c \in \mathbb{R}^k$, let

$$\rho_\lambda(c) = \lambda c^T E c + \frac{1}{n}\sum_{i=1}^{n}[\lambda_i s_c - z_i]^2. \tag{5}$$

The second term in (5) is just the mean square error in approximating the data by s_c. The first term can be thought of as a *penalty term* which, with the appropriate choice of E, measures the smoothness of s_c. The parameter λ can be regarded as a *smoothing* parameter.

Definition 1. *The penalized least-squares fit of the function f based on data z is the function $s_{\lambda,z} := s_{c(\lambda)}$ corresponding to the choice of $c(\lambda)$ which minimizes the functional $\rho_\lambda(c)$ defined in (5).*

While the use of such penalty terms seems to be well established practice in statistical data fitting as well as in regularization of ill-posed problems, the method seems to have received much less attention in Approximation Theory circles. For more on the history of penalized least squares methods, see Remark 1 in Section 6 below.

The remainder of this paper is organized as follows. In the following section we establish the existence and uniqueness of solutions to the penalized least squares problem, and also discuss the special cases when $\lambda = 0$ and when λ approaches ∞. In Section 3 we discuss the connection between the choice of the smoothing parameter λ and the goodness of fit. Section 4 is devoted to an automatic procedure for selecting the smoothing parameter λ called *generalized cross validation*. In Section 5 we give details of how penalized least squares can be applied to fit data in the univariate case using splines. The last section is devoted to remarks.

2. Existence and uniqueness

Throughout the remainder of this paper we assume that the linear functionals $\lambda_1, \ldots, \lambda_n$ and basis functions $B_1, \ldots, B_k$ are such that for some $1 \le \nu_1 < \cdots < \nu_k \le n$,

$$\det\, (\lambda_{\nu_i} B_j)_{i,j=1}^{k} \neq 0. \tag{6}$$

Under this basic hypothesis, we can now establish the following existence and uniqueness theorem for penalized least-squares fitting.

Theorem 2. *For any $\lambda \ge 0$, there exists a unique vector $c(\lambda)$ minimizing the functional $\rho_\lambda(c)$ in (5). In particular, $c(\lambda)$ is the unique solution of the system*

$$(B^T B + n\lambda E)c(\lambda) = B^T z, \tag{7}$$

where $z = (z_1, \ldots, z_n)^T$ and B is the $n \times k$ matrix with entries

$$B_{ij} = \lambda_i B_j, \quad i = 1, \ldots, n; \; j = 1, \ldots, k.$$

Proof: Setting the gradient of the function $\rho_\lambda(c)$ equal to 0 leads immediately to the system of equations (7). The basic hypothesis (6) assures that the matrix $G := B^T B$ is symmetric, positive definite, and nonsingular. Since E is assumed to be symmetric nonnegative-definite, we conclude that for every $\lambda \ge 0$, $G + n\lambda E$ is also symmetric positive definite. ■

As in the case of the classical smoothing splines, clearly the size of the smoothing parameter λ has a major impact on how well the function $s_{\lambda,z}$ fits the function f which generated the data. In Sections 3 and 4 below we shall discuss how to select λ effectively. In the remainder of this section we discuss the nature of the fit in the extreme cases when $\lambda = 0$ and when λ approaches ∞.

When $\lambda = 0$, it is clear that the functional $\rho_\lambda(c)$ is just the mean square error when fitting the data using functions of the form (2). Hence this case corresponds to classical least-squares fitting using the space $\mathcal{S} = \text{span } \{B_i\}_1^k$, and the system (7) is just the usual set of normal equations. If $k = n$, then the least-squares fit is simply the function in $\mathcal{S}$ which interpolates the data.

The situation as λ approaches ∞ is somewhat more complicated. First we prove a lemma concerning the behavior of the inverse of the matrix $G + n\lambda E$, where $G = B^T B$.

Lemma 3. *For all $\lambda \ge 0$, the entries of $(G + n\lambda E)^{-1}$ are uniformly bounded by $1/\alpha$, where α is the smallest eigenvalue of G.*

Proof: Let x be the r^{th} column of $(G + n\lambda E)^{-1}$. Then $(G + n\lambda E)x = e_r$, where e_r is the vector in R^k with all zero entries except for the r^{th}. It follows that $x^T(G + n\lambda E)x = x^T e_r = x_r$. Now by the nonnegative definiteness of E, we conclude that $x^T G x \le x_r$. On the other hand, by the positive definiteness of G, the smallest eigenvalue α of G satisfies

$$0 < \alpha = \min_{y \neq 0} \frac{y^T G y}{y^T y}.$$

It follows that $\alpha x^T x \leq x^T G x \leq x_r$, and thus $\alpha x_j^2 \leq x_r$, for all $j = 1, \ldots, k$. This inequality asserts in particular that $0 \leq x_r \leq 1/\alpha$, and it immediately follows that $|x_j| \leq 1/\alpha$, for all $j = 1, \ldots, k$. Since r was an arbitrary integer satisfying $1 \leq r \leq k$, this completes the proof. ■

We can now examine the behavior as $\lambda \to \infty$ of the coefficient vector $c(\lambda)$ which minimizes the functional $\rho_\lambda(c)$ in (5).

Lemma 4. *There exists a vector $c(\infty)$ in R^k such that*

$$\lim_{\lambda \to \infty} c(\lambda) = c(\infty).$$

Moreover, $Ec(\infty) = 0$.

Proof: Clearly, $c(\lambda) = (G + n\lambda E)^{-1} B^T z$. Since the entries of $(G + n\lambda E)^{-1}$ are rational functionals of λ and are uniformly bounded for all $\lambda \geq 0$, it follows that $(G + n\lambda E)^{-1}$ converges to some matrix L as $\lambda \to \infty$. But then $c(\lambda)$ approaches $LB^T z$ as $\lambda \to \infty$. Finally, since

$$\left(\frac{G}{n\lambda} + E \right) c(\lambda) = \frac{B^T z}{n\lambda},$$

it follows that $Ec(\infty) = 0$. ■

Theorem 5. *The function $s_{c(\infty)}$ is the least-squares fit of the data from the following subspace of $\mathcal{S}$:*

$$\mathcal{S}_E = \left\{ \sum_{i=1}^{k} c_i B_i \ : \ Ec = 0 \right\}. \tag{8}$$

Proof: Since

$$Gc(\lambda) + n\lambda Ec(\lambda) = B^T z,$$

and the first term converges as $\lambda \to \infty$, we conclude that $n\lambda Ec(\lambda)$ converges to $u := B^T z - Gc(\infty)$. The matrices E and (E, u) have the same rank, and it follows that $u = Ex$ for some element Ex with x in R^k. Thus we have

$$\begin{bmatrix} G & E \\ E & 0 \end{bmatrix} \begin{bmatrix} c(\infty) \\ x \end{bmatrix} = \begin{bmatrix} B^T z \\ 0 \end{bmatrix}.$$

But this is precisely the system of equations which uniquely determines the least-squares fit of the data from the subspace $\mathcal{S}_E$. ■

3. Properties of the fit

In this section we discuss the behavior of the fit as we vary the smoothing parameter λ. Given measurements as in (1), let $s_{\lambda,z}$ be the penalized least-squares fit of f constructed from the noisy data vector z. Then the mean-square error using this fit is given by

$$T_z(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[\lambda_i(s_{\lambda,z} - f)]^2. \tag{9}$$

We begin by considering the case where we have exact measurements; i.e., where the errors $\epsilon_1, \ldots, \epsilon_n$ in (1) are all zero. In this case we want to consider $T_{\mathbf{f}}(\lambda)$, where $\mathbf{f} = (\lambda_1 f, \ldots, \lambda_n f)^T$.

Theorem 6. *The function $T_{\mathbf{f}}(\lambda)$ is monotone increasing for $\lambda \geq 0$ with $\dot{T}_{\mathbf{f}}(0) = 0$ and $\lim_{\lambda\to\infty} \dot{T}_{\mathbf{f}}(\lambda) = 0$.*

Proof: First we establish the monotonicity of $T_{\mathbf{f}}(\lambda)$. For exact measurements $z = \mathbf{f}$, the minimum of the expression ρ_λ in (5) is attained at $c_{\mathbf{f}}(\lambda)$, and is given by

$$\rho_\lambda(c_{\mathbf{f}}(\lambda)) = \lambda c_{\mathbf{f}}(\lambda)^T E c_{\mathbf{f}}(\lambda) + T_{\mathbf{f}}(\lambda).$$

Similarly, for all $\tilde{\lambda} > \lambda > 0$,

$$\rho_{\tilde{\lambda}}(c_{\mathbf{f}}(\tilde{\lambda})) = \tilde{\lambda} c_{\mathbf{f}}(\tilde{\lambda})^T E c_{\mathbf{f}}(\tilde{\lambda}) + T_{\mathbf{f}}(\tilde{\lambda}).$$

For ease of notation, we now define $\alpha = c_{\mathbf{f}}(\lambda)^T E c_{\mathbf{f}}(\lambda)$, $\tilde{\alpha} = c_{\mathbf{f}}(\tilde{\lambda})^T E c_{\mathbf{f}}(\tilde{\lambda})$, $\beta = T_{\mathbf{f}}(\lambda)$, and $\tilde{\beta} = T_{\mathbf{f}}(\tilde{\lambda})$. Then the minimality of $c_{\mathbf{f}}(\lambda)$ and $c_{\mathbf{f}}(\tilde{\lambda})$, respectively, imply that

$$\lambda\alpha + \beta \leq \lambda\tilde{\alpha} + \tilde{\beta} \quad \text{and} \quad \tilde{\lambda}\tilde{\alpha} + \tilde{\beta} \leq \tilde{\lambda}\alpha + \beta.$$

It follows that

$$\lambda\alpha + \tilde{\lambda}\tilde{\alpha} \leq \lambda\tilde{\alpha} + \tilde{\lambda}\alpha \quad \text{and} \quad \tilde{\lambda}(\tilde{\alpha} - \alpha) \leq \lambda(\tilde{\alpha} - \alpha),$$

and thus that $\tilde{\alpha} \leq \alpha$. Since $\lambda(\alpha - \tilde{\alpha}) \leq \tilde{\beta} - \beta$, we conclude that $\tilde{\beta} \geq \beta$ which asserts that $T_{\mathbf{f}}(\lambda)$ is monotone increasing.

To establish the assertions about the derivative of $T_{\mathbf{f}}(\lambda)$, we need an explict formula for it. To this end we introduce the $n \times n$ *influence matrix*

$$A(\lambda) = B(G + n\lambda E)^{-1} B^T. \tag{10}$$

By the properties of the matrices appearing in (10), it follows immediately that $A(\lambda)$ is symmetric positive definite. In addition, we claim that $A(\lambda)$ provides the connection between the data vector z, and the associated values of the penalized least squares fit; i.e.,

$$\begin{bmatrix} \lambda_1 s_{\lambda,z} \\ \vdots \\ \lambda_n s_{\lambda,z} \end{bmatrix} = A(\lambda) z. \tag{11}$$

Indeed, $(\lambda_1 s_{\lambda,z},\dots,\lambda_n s_{\lambda,z})^T = Bc(\lambda) = B(G+n\lambda E)^{-1}B^T z = A(\lambda)z$.

Now it is clear that

$$T_{\mathbf{f}}(\lambda) = \frac{1}{n}[A(\lambda)\mathbf{f}-\mathbf{f}]^T[A(\lambda)\mathbf{f}-\mathbf{f}].$$

From this we compute that

$$n[T_{\mathbf{f}}(\lambda)-T_{\mathbf{f}}(0)] = \mathbf{f}^T[A(\lambda)^T A(\lambda) - 2A(\lambda) - A(0)^T A(0) + 2A(0)]\mathbf{f}.$$

Using (10), we arrrive at the formula

$$T_{\mathbf{f}}(\lambda) = T_{\mathbf{f}}(0) + n\lambda^2 c_{\mathbf{f}}(\lambda)^T E G^{-1} E c_{\mathbf{f}}(\lambda). \tag{12}$$

Forming the difference quotient of $T_{\mathbf{f}}(\lambda)$ at 0 and using the continuity of $c_{\mathbf{f}}(\lambda)$, (as shown in the proof of Lemma 4), it immediately follows that the derivative at $\lambda = 0$ is zero as asserted. The assertion about the derivative as $\lambda \to \infty$ follows from the boundedness of $T_{\mathbf{f}}(\lambda)$. ■

It is now clear that to minimize the error in the approximation when fitting exact data, we should choose the value of the smoothing parameter λ to be zero. We should note, however, that for a certain class of functions $\mathcal{F}_E$ which we are about to define, $T_{\mathbf{f}}(\lambda)$ is constant, and hence all values of λ are equally as good. The class is

$$\mathcal{F}_E = \{f \ : \ d(f,\mathcal{S}) = d(f,\mathcal{S}_E)\}, \tag{13}$$

where

$$d(f,\mathcal{S}) = \inf_{s\in\mathcal{S}} \frac{1}{n}\sum_{i=1}^{n}[\lambda_i(s-f)]^2, \tag{14}$$

and $d(f,\mathcal{S}_E)$ is defined similarly. Indeed, in this case, $T_{\mathbf{f}}(\lambda)$ has the same value for both $\lambda = 0$ and for $\lambda = \infty$. Thus, since $T_{\mathbf{f}}(\lambda)$ is monotone increasing, it must be a constant.

In practice we almost always have noisy measurements, and as we shall see shortly, in this case it will be more advantageous to take $\lambda > 0$. To analyze the noisy case, we need to make some basic assumption about the nature of the noise. Throughout the remainder of this section we suppose that the errors $\epsilon_1,\dots,\epsilon_n$ in the measurements (1) are independent, identically distributed random variables with mean 0 and variance σ^2.

It is easy to see that

$$nT_z(\lambda) = nT_{\mathbf{f}}(\lambda) + 2\epsilon^T A(\lambda)^T[A(\lambda)\mathbf{f}-\mathbf{f}] + \epsilon^T A(\lambda)^T A(\lambda)\epsilon.$$

Taking the *expected value* of this expression gives

$$\mathcal{E}T_z(\lambda) = T_{\mathbf{f}}(\lambda) + \frac{\sigma^2 \text{trace}\,(A^2(\lambda))}{n}. \tag{15}$$

Before analyzing this expression further, we need some information on the trace of the influence matrix $A(\lambda)$. Let $0 \le \kappa_1 \le \kappa_2 \le \cdots \le \kappa_k$ be the eigenvalues of $G^{-1}E$, and suppose that $\{v_1,\dots,v_k\}$ are the corresponding eigenvectors. We suppose that $\kappa_1 = \cdots = \kappa_d = 0 < \kappa_{d+1}$. This is equivalent to assuming that the dimension of the space $\mathcal{S}_E$ defined in (8) is d.

Lemma 7. *For $0 \leq \lambda < \infty$ the matrix $A(\lambda)$ has exactly k non-zero eigenvalues*

$$\mu_j = \frac{1}{1 + n\lambda\kappa_j}, \quad j = 1, \ldots, k.$$

For $\lambda = \infty$, the eigenvalues of $A(\lambda)$ consist of 0 with multiplicity $n - d$ and 1 with multiplicity d.

Proof: We first consider the case where $0 \leq \lambda < \infty$. Since for all z, the vector $A(\lambda)z$ lies in a k-dimensional subspace of $\mathbb{R}^n$, the first $n - k$ eigenvalues of A are all 0. Suppose now that $\lambda = 0$, and fix $1 \leq j \leq k$. Since the least squares fit of the function B_j is B_j itself, it follows that $(\lambda_1 B_j, \ldots, \lambda_n B_j)^T$ is an eigenvector of $A(0)$ corresponding to eigenvalue 1. This proves that the k nonzero eigenvalues of $A(0)$ are all 1.

We now treat the case where $0 < \lambda < \infty$. Suppose $v \in \mathbb{R}^k$ is an eigenvector of $G^{-1}E$ corresponding to $\kappa > 0$; i.e., $G^{-1}Ev = \kappa v$. Let $\mu = \frac{1}{1+n\lambda\kappa}$ and suppose $u \in \mathbb{R}^n$ is such that $G^{-1}B^T u = v$. Let $w = Au$. Since B is of rank k,

$$Au = 0 \qquad \text{iff} \qquad B^T u = 0. \tag{16}$$

The fact that $v \neq 0$ coupled with (16) assures that $w \neq 0$. Now it is easy to see that the following equalities are equivalent:

$$\kappa v = G^{-1}Ev.$$

$$Gv = \mu Gv + n\lambda\mu Ev$$

$$Gv = \mu(G + n\lambda E)v$$

$$(G + n\lambda E)^{-1}B^T u = \mu G^{-1}B^T u$$

$$G(G + n\lambda E)^{-1}B^T u = \mu B^T u$$

$$B^T(Au - \mu u) = 0$$

$$A(Au - \mu u) = 0$$

$$Aw = \mu w.$$

This establishes that μ is an eigenvector of A. To complete the proof we have to consider the possibility that κ is an eigenvalue of $G^{-1}E$ of multiplicity m. Suppose $v_1, \ldots, v_m$ are linear independent eigenvectors of $G^{-1}E$ associated with κ. For each $i = 1, \ldots, m$, let $u_i \in \mathbb{R}^n$ be such that $G^{-1}B^T u_i = v_i$, and let $w_i = Au_i$. Now we claim that the w_i are linearly independent eigenvectors of A. Indeed, if $\alpha_1 w_1 + \cdots + \alpha_m w_m = 0$, then $\alpha_1 Au_1 + \cdots + \alpha_m Au_m = 0$, and by (16), it follows that $\alpha_1 G^{-1}B^T u_1 + \cdots + \alpha_m G^{-1}B^T u_m = \alpha_1 v_1 + \cdots + \alpha_m v_m = 0$. But by the linear independence of the v's, this implies $\alpha_1 = \cdots = \alpha_m = 0$, and we have established the asserted linear independence of the w_i's.

It remains to deal with the case $\lambda = \infty$. In this case, we are performing least squares in $\mathcal{S}_E$, and hence s_λ lies in a d dimensional subspace of $\mathbb{R}^n$. It follows that $n - d$ of the eigenvalues of $A(\infty)$ must be zero. On the other hand, since each of the d basis functions of $\mathcal{S}_E$ is approximated exactly, just as in the case $\lambda = 0$ we see that 1 is an eigenvalue of $A(\infty)$ of multiplicity d. ■

For later use, we note that since the trace of a matrix is equal to the sum of its eigenvalues, for $0 \leq \lambda < \infty$,

$$t(\lambda) := \text{trace } A(\lambda) = \sum_{i=1}^{k} \left(\frac{1}{1 + n\lambda\kappa_i} \right), \tag{17}$$

and, since the eigenvalues of $A^2(\lambda)$ are the squares of the eigenvalues of $A(\lambda)$,

$$t_2(\lambda) := \text{trace } A^2(\lambda) = \sum_{i=1}^{k} \left(\frac{1}{1 + n\lambda\kappa_i} \right)^2. \tag{18}$$

In particular, $t(0) = t_2(0) = k$. We also note that $t(\infty) = t_2(\infty) = d$.

Theorem 8. *The function $\phi(\lambda) := \mathcal{E}T_z(\lambda)$ has the value $\mathcal{E}T_z(0) = T_f(0) + k\sigma^2/n$, and asymptotically approaches the value $T_f(\infty) + d\sigma^2/n$ as $\lambda \to \infty$. Its derivative at $\lambda = 0$ is negative. Moreover, for all $\lambda \geq 0$,*

$$\mathcal{E}T_z(\lambda) - \mathcal{E}T_z(0) \geq \frac{\sigma^2}{n}[t_2(\lambda) - t_2(0)], \tag{19}$$

where $t_2(\lambda)$ is defined in (18).

Proof: Putting $t_2(0) = k$ and $t_2(\infty) = d$ in (15), we immediately obtain the asserted values of ϕ at $\lambda = 0$ and $\lambda = \infty$. By (15),

$$\dot{\phi}(\lambda) = \dot{T}_f(\lambda) + \frac{\sigma^2 \dot{t}_2(\lambda)}{n}. \tag{20}$$

Now since

$$\dot{t}_2(\lambda) = -2 \sum_{i=1}^{k} \frac{n\kappa_i}{(1 + n\lambda\kappa_i)^3}$$

and $\dot{T}_f(0) = 0$ by Theorem 6, we immediately deduce that $\dot{\phi}(0) < 0$ as asserted.

Finally, to prove (19), we note that by (12) and (15),

$$\mathcal{E}T_z(\lambda) - \mathcal{E}T_z(0) = n\lambda^2 c_f(\lambda)^T E G^{-1} E c_f(\lambda) + \frac{\sigma^2[t_2(\lambda) - t_2(0)]}{n}.$$

Since the first term on the right is nonnegative, the result follows. ■

It is clear from (20) that in the case where $f \in \mathcal{F}_E$, the function $\mathcal{E}T_z(\lambda)$ is monotone decreasing since, as observed earlier, $T_f(\lambda)$ is constant in this case while $\dot{t}_2(\lambda) < 0$.

4. Choosing the smoothing parameter λ

In the previous section, we have examined the connection between the choice of the smoothing parameter and the size of the true error or expected value of the true error. We can summarize our observations as follows:

Case 1: (The noise is zero) In this case, in order to minimize the true error of fit, we should take $\lambda = 0$; i.e., we should perform least squares using the space $\mathcal{S}$.

Case 2: (Noisy data) In this case it is reasonable to choose λ to minimize the expected value $\mathcal{E}T_z(\lambda)$ of the true error. There are two subcases. If $\mathcal{E}T_z(\lambda)$ is monotone decreasing on $[0,\infty)$, then we should take $\lambda = \infty$. On the other hand, if $\mathcal{E}T_z(\lambda)$ is not monotone decreasing on $[0,\infty)$, then since by Theorem 8, $\mathcal{E}T_z(\lambda)$ has a negative derivative at $\lambda = 0$, there is at least one value of λ in $(0,\infty)$ which minimizes $\mathcal{E}T_z(\lambda)$. Let λ_n^T be the smallest choice of λ which provides a minimum.

In the remainder of this section we restrict our attention to the case of noisy data. Our immediate aim is to discuss an automatic method, called *generalized cross validation*, for computing a reasonable estimate for λ_n^T. Given a data vector z as in (1), let $s_{\lambda,z}$ be the associated penalized least-squares fit, and let $A(\lambda)$ be the influence matrix defined in (10). Then we define the associated *generalized cross validation function* as

$$V(\lambda) = \frac{E_z(\lambda)}{[1 - t(\lambda)/n]^2}, \tag{21}$$

where

$$E_z(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(\lambda_i s_{\lambda,z} - z_i)^2, \tag{22}$$

and $t(\lambda)$ is the trace of $A(\lambda)$ as given in (17).

Lemma 9. *The function $E_z(\lambda)$ is monotone increasing for $\lambda \geq 0$ with $\dot{E}_z(0) = 0$ and $\lim_{\lambda\to\infty} \dot{E}_z(\lambda) = 0$.*

Proof: The proof of this lemma is virtually the same as the proof of Theorem 6, and we omit the details. ■

Lemma 9 does not assert that $E_z(\lambda)$ is *strictly* monotone increasing. For some z it can happen that $E_z(\lambda)$ is constant for all $0 \leq \lambda < \infty$. We now show that $V(\lambda)$ has properties very similar to those of $\mathcal{E}T_z(\lambda)$.

Theorem 10. *The function $V(\lambda)$ has value $E_z(0)/[1 - k/n]^2$ at 0 and asymptotically approaches the value $E_z(\infty)/[1 - d/n]^2$ as $\lambda \to \infty$. Moreover, $V(\lambda)$ has a negative derivative at $\lambda = 0$.*

Proof: The first assertions follow immediately from the properties of $t(\lambda)$. To prove the last assertion, we note that

$$\dot{V}(\lambda) = \frac{n^2(n - t(\lambda))\dot{E}_z(\lambda) + 2n^2 E_z(\lambda)\dot{t}(\lambda)}{[n - t(\lambda)]^3}.$$

The assertions now follow from the properties of $E_z(\lambda)$ and the fact that

$$\dot{t}(\lambda) = -\sum_{i=1}^{k} \frac{n\kappa_i}{(1+n\lambda\kappa_i)^2}. \quad \blacksquare$$

Theorem 10 implies that either $V(\lambda)$ is monotone decreasing on $[0,\infty)$ in which case we can think of its minimum as occuring at $\lambda_n^V := \infty$, or there is at least one value of λ in $(0,\infty)$ where E_z has an absolute minimum, in which case we define λ_n^V to be the smallest choice of λ which works.

The process of computing the quantity λ_n^V as an approximation to λ_n^T is called the *method of generalized cross validation.* To implement it numerically, we have to find the minimum of the validation function $V(\lambda)$. The standard approach to doing this is to compute $V(\lambda)$ at several choices of λ, and then use a search procedure. For each choice of λ, the bulk of the computational effort typically goes into finding $t(\lambda)$ = trace $A(\lambda)$. In those cases where the matrix B has a band structure (as happens for example when using univariate splines – see Section 5 below), the trace can be computed efficiently using the LU decomposition of the matrix $(G+n\lambda E)$, see [13,21]. In the general case it may be more efficient to resort to finding the singular value decomposition of $(G+n\lambda E)$, which then leads to an explicit expression for $t(\lambda)$, see [2].

From a theoretical standpoint, it is of interest to consider the expected value of the validation function V. Since

$$\mathcal{E}E_z(\lambda) = T_f(\lambda) + \frac{\sigma^2}{n} t_3(\lambda),$$

where

$$t_3(\lambda) = \text{trace } (A-I)^2,$$

we obtain

$$\mathcal{E}V(\lambda) = \frac{T_f(\lambda) + \frac{\sigma^2 t_3(\lambda)}{n}}{[1-\frac{t(\lambda)}{n}]^2}.$$

The following theorem, whose proof is straightforward, shows that $\mathcal{E}V(\lambda)$ has properties very similar to those of both $\mathcal{E}T_z(\lambda)$ and $V(\lambda)$.

Theorem 11. *The function $\mathcal{E}V(\lambda)$ has the value $\frac{T_f(0)}{[1-\frac{k}{n}]^2}$ at 0, and asymptotically approaches the value $\frac{T_f(\infty)}{[1-\frac{d}{n}]^2}$ as $\lambda \to \infty$. Its derivative at $\lambda = 0$ is negative.*

Theorem 11 shows that either the function $\mathcal{E}V(\lambda)$ is monotone decreasing for all λ, or there is some value of $0 < \lambda < \infty$ where it has a minimum. We denote the first such point by λ_n^{EV}. In practice, we cannot compute this value because we have no way of computing either $T_f(\lambda)$ or σ. (Indeed, if we could compute these quantities, we could find the minimum of $\mathcal{E}T_z(\lambda)$ directly). We have the following interesting theorem concerning λ_n^{EV}.

Theorem 12. *The quantity λ_n^{EV} is an asymptotically optimal estimate of the best smoothing parameter λ_n^T in the sense that*

$$1 \leq \frac{\mathcal{E}T_z(\lambda_n^{EV})}{\mathcal{E}T_z(\lambda_n^T)} \to 1 \tag{23}$$

as $n \to \infty$.

Proof: As in [23], it is easy to show that for $\lambda > 0$,

$$1 \leq \frac{\mathcal{E}T_z(\lambda_n^{EV})}{\mathcal{E}T_z(\lambda_n^T)} \leq \frac{1+\Delta_n(\lambda_n^T)}{1-\Delta_n(\lambda_n^{EV})}, \tag{24}$$

where

$$\Delta_n(\lambda) = \frac{\left|t(\lambda)^2 - 2nt(\lambda) + nt(\lambda)^2/t_2(\lambda)\right|}{[n-t(\lambda)]^2},$$

and where $t(\lambda)$ and $t_2(\lambda)$ are defined in (17) and (18), respectively. Now since $d \leq t(\lambda) \leq k$ and $d \leq t_2(\lambda) \leq k$, it follows that the quotient in (24) converges to zero at a rate $\mathcal{O}(\frac{1}{n})$ as $n \to \infty$. ■

For smoothing splines, the only known way of producing sequences λ_n which are asymptotically optimal is via generalized cross validation. The following result shows, however, that for penalized least squares fitting, the situation is different. Here there is a range of values of λ_n which are asymptotically optimal.

Theorem 13. *Suppose that for each $n > 1$, the quantity $\lambda_n \geq 0$ is such that $\mathcal{E}T_z(\lambda_n) \leq \mathcal{E}T_z(0)$. Then*

$$\lim_{n\to\infty} \frac{\mathcal{E}T_z(\lambda_n)}{\mathcal{E}T_z(\lambda_n^T)} = 1. \tag{25}$$

Proof: By (19) and the fact that $t_2(0) - t_2(\lambda) \leq k - d$, we have

$$1 \leq \frac{\mathcal{E}T_z(\lambda_n)}{\mathcal{E}T_z(\lambda_n^T)} \leq \frac{\mathcal{E}T_z(0)}{\mathcal{E}T_z(0) - (k-d)\sigma^2/n}.$$

Clearly this expression converges to 1 at a rate of $\mathcal{O}(\frac{1}{n})$. ■

Theorem 13 asserts in particular that $\lambda_n = 0$ provides an asymptotically optimal sequence of smoothing parameters. In fact, since $\mathcal{E}T_z(\lambda)$ has a negative derivative at $\lambda = 0$, any sequence of "sufficiently small" numbers λ_n would work.

The fact that $\mathcal{E}T_z(0) - (k-d)\sigma^2/n \leq \mathcal{E}T_z(\lambda) \leq \mathcal{E}T_z(0)$ for all sufficiently small λ suggests that for fairly large values of n, there is little to gain in terms of goodness of fit by striving for the optimal value of λ; it is enough to make sure that λ is sufficiently small. On the other hand, for noisy data, in practice it is often necessary to take $\lambda > 0$ in order to get a smooth fit. Generalized cross validation can be a useful automatic procedure for selecting a good value of λ.

5. Penalized Least Squares Fitting of Curves Using Splines

In this section we show how the penalized least squares method can be applied to fit a spline to the data

$$\lambda_i f = f(t_i) + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $f \in C[a,b]$, and

$$a = t_1 < \cdots < t_n = b.$$

Let $m, k > 0$ be prescribed integers, and suppose that

$$B_i(x) = N_i^{2m}(x), \tag{26}$$

are the normalized B-splines of order m associated with an extended knot vector $y_1 \le \cdots \le y_{2m+k}$, with $y_i < y_{i+2m}$ for all i (cf. [20]). We may assume that the knots have been chosen in such a way that $y_{2m} \le a$ and $b \le y_{k+1}$, and so that (6) is satisfied. By the Schoenberg-Whitney Theorem (cf. Theorem 4.64 of [20]), (6) will be satisfied provided we choose the knots so that there exist $1 < \nu_2 < \cdots < \nu_{k-1} < n$ so that $y_i < t_{\nu_i} < y_{i+m}$ for all $i = 2, \ldots, k-1$.

When using splines, a natural measure for the smoothness of a fit s would be

$$J(s) = \int_a^b [D^m g(t)]^2 dt. \tag{27}$$

In this case, if s is a spline of the form

$$s(x) = \sum_{i=1}^k c_i B_i(x),$$

then $J(g) = c^T E c$ with

$$E_{ij} = \int_a^b D^m B_i(t) D^m B_j(t)\, dt, \quad i,j = 1, \ldots, k. \tag{28}$$

We now discuss the problem of computing the quantities appearing in (28). First we show how to represent the derivative of a B-spline in terms of lower order ones.

Lemma 14. *Let $N_i^\ell(x)$ be the normalized B-splines of order ℓ associated with the knot sequence $y_i \le \cdots \le y_{i+\ell}$ with $y_i < y_{i+\ell}$ (cf. [20]). Define $h_{i,\nu} = y_{i+\nu} - y_i$. Then for any $q > 0$,*

$$D^q N_i^\ell(x) = \sum_{\nu=0}^q a_{i,\nu}^q Q_{i+\nu}^{\ell-q}, \tag{29}$$

where the Q's are unnormalized B-splines on the same knots, and where the $a^q_{i,\nu}$'s can be computed recursively as follows:

1. Set
$$A_\nu = \begin{cases} 1, & \nu = 2 \\ -1, & \nu = 3 \\ 0, & \nu = 1, 4, \dots q+2. \end{cases}$$

2. Do $\nu = 2$ to q

    Do $\mu = \nu + 2$ step -1 to 2
    $A_\mu = (\frac{A_\mu}{h_{i+\mu-2,\ell+1-\nu}} - \frac{A_{\mu-1}}{h_{i+\mu-3,\ell+1-\nu}})$

3. Do $\nu = 0$ to q
    $a_{i,\nu} = fac * A_{\nu+2}$,

where fac $= (\ell - 1)(\ell - 2) \cdots (\ell - q)$. If any of the h's is zero in step 2, the corresponding term should be omitted.

Proof: This result follows easily from the basic formula (cf. Theorem 4.16 in [20])

$$DQ^\ell_i(x) = (\ell - 1)\frac{(Q^{\ell-1}_i(x) - Q^{\ell-1}_{i+1}(x)}{h_{i,\ell}},$$

and the fact that $N^\ell_i(x) = h_{i,\ell}Q^\ell_i(x)$. ■

Theorem 15. *For all i, j we have*

$$E_{i,j} = \int_a^b D^m Q^{2m}_i(x) D^m Q^{2m}_j(x)dx = \sum_{\nu=0}^{m}\sum_{\mu=0}^{m} a^m_{i,\nu} a^m_{i,\mu} \int_a^b Q^m_{i+\nu}(x) Q^m_{i+\mu}(x)dx.$$

Proof: The result follows easily from Lemma 14. ■

Theorem 15 can easily be converted into an algorithm for computing the entries of the matrix E. Let

$$P = \left(\int_a^b Q^m_i(x) Q^m_j(x)dx\right)^k_{i,j=1}. \tag{30}$$

This is a $2m - 1$ banded symmetric matrix which can be computed accurately and efficiently using Gauss quadrature (see [4,20]). Then

$$E_{i,j} = a^T_i P(i,j) a_j,$$

where $P(i,j)$ is the $(m+1) \times (m+1)$ minor of P obtained by taking rows $i, \dots, i+m$ and columns $j, \dots, j+m$, and where $a_i = [a^m_{i,0}, \dots, a^m_{i,m}]^T$ for all i.

Penalized least squares using cubic splines with uniform knots has been used in [11] to fit some medical data, and in [15] to fit some mechanical data.

6. Remarks

Remark 1

The idea of minimizing a combination of goodness of fit and some measure of smoothness is well-known in the approximation literature. Such expressions arise, for example, in the definition of smoothing splines (see [5,13-14,16-25,31]) and of thin plate splines (see [7-10,25,28-30]). In these methods the approximating function turns out to be a linear combination of n basis functions (see Remark 3 below). The idea of working with a smaller number k of basis functions was explicitly mentioned in Wahba [26-28], but seems to have been carried out explicitly in only a few papers. In curve fitting case, cubic splines with equally spaced knots were used in [11] to fit some medical data, and in [15] to fit some mechanical data. Penalized least squares methods have also been used to fit surfaces to scattered data using tensor-product splines [6] and finite elements defined on rectangles [1].

Remark 2

The method of penalized least squares as described here is closely related to the method of ridge regression as studied by statisticians (see, for example, [12]). The ridge regression problem is to minimize the expression $\rho_\lambda(c)$ in (5), with E replaced by the identity matrix. It has been shown in [2] that the general penalized least squares problem can be reduced to the ridge regression problem by appropriate matrix manipulations.

Remark 3

Suppose $a = t_1 < \cdots < t_n = b$ as in Section 5. Then if we minimize

$$\lambda \int_a^b [f^m(x)]^2 dx + \frac{1}{n} \sum_{i=1}^{n} [f(t_i) - z_i]^2$$

over all functions with square integrable m-th derivative, we get the classical natural smoothing spline (see [5,13-14,16-25,31]). The theory of smoothing splines can be regarded as a special case of the penalized least squares method treated here. Indeed, if we set $k = n$ and take the natural B-splines (cf. [14,20]) as basis functions, then the solution of the penalized least squares problem is precisely the smoothing spline. Almost all of what we have done here is valid for the case where $k = n$ with the exception of the assertions about asymptotic optimality (see Remark 4).

Remark 4

The asymptotic assertion of Theorems 12 and 13 have been proved here only for the case where k is fixed. However. it is not hard to see that the results are also valid for a sequence $k(n)$ with the property that $k(n)/n \to 0$ as $n \to \infty$. The situation when $k(n)/n$ does not go to zero as $n \to \infty$ is more delicate. For example, to prove the analog of Theorem 12 for natural smoothing splines (see [22]) or for complete smoothing splines (cf. [21]), it was necessary to give precise estimates on the eigenvalues of E.

Remark 5

In Section 5 we showed how to use univariate splines to fit noisy data in one dimension by penalized least squares. Clearly, the method can also be used to fit surfaces to noisy data. For example, suppose f is defined on some subset $\Omega \subseteq \mathbb{R}^2$. Then we may measure the smoothness of possible fits g of f by

$$J(g) = (g, g)_m, \tag{31}$$

where, in general,

$$(\phi, \psi)_m := \sum_{\nu+\mu=m} \int D_x^\nu D_y^\mu \phi(x,y) D_x^\nu D_y^\mu \psi(x,y)\, dx dy. \tag{32}$$

In this case, if s is as in (2), then $J(s) = c^T E c$ with

$$E_{ij} = (B_i, B_j)_m.$$

Penalized least squares of surfaces can be performed using a variety of bases including tensor product splines (cf. [6]), radial basis functions (cf. [26]), finite elements (cf. [1]), or multivariate splines on triangulations.

The energy expression (31) arises in the definition of thin plate splines (see [7-10,25,28-30] and references therein). Thin plate splines can be regarded as arising from penalized least squares using $k = n$ and appropriate basis functions.

Remark 6

Generalized cross validation methods have been heavily studied in the case of spline smoothing (cf. [5,13-14,16-25,31]) as well as for ridge regression [12] and for regularization of ill-posed problems [27]. In connection with general penalized least squares problems, generalized cross validation is treated in [2]. It was also used in [15] in connection with a cubic spline method. The basic statistical hypothesis that the errors are independent identically distributed random variables is standard in discussing such methods. We should note that we are *not* assuming that the distributions are Gaussian, only that they are all the same. However, even this hypothesis may well not be satisfied in some practical situations.

Remark 7

Error bounds for spline smoothing can be found in [16,19] and for thin plate splines in [25]. Error bounds for penalized least squares fitting of surfaces using finite elements can be found in [1]. We intend to treat error bounds for penalized least squares in a separate paper.

Remark 8

In some practical problems, in forming the error expression (3), it may be desirable to weigh the i-th measurement with a weight factor $w_i > 0$ for $i = 1, \ldots, n$. In this case, the results of the paper remain valid with only minor adjustments.

Acknowledgment: The work of the second author was supported in part by the Deutsche Forschungsgemeinschaft, and by the National Science Foundation under Grant DMS-8602337.

References

1. Apprato, D., R. Arcangeli, and R. Manzanilla, On the construction of surfaces of class C^k, preprint, 1987.
2. Bates, D. M., M. J. Lindstrom, G. Wahba, and B. S. Zandell, GCVPACK–Routines for generalized cross validation, Tech. Rpt. 775, Statistics Dept., Univ. Wisconsin, Madison, 1986.
3. Bates, D. and G. Wahba, Computational methods for generalized cross-validation with large data sets, in *Treatment of Integral Equations by Numerical Methods*, C. T. Baker and G. Miller (eds.), Academic Press, New York, 1982, 283–296.
4. de Boor, C., T. Lyche, and L. L. Schumaker, On calculating with B-splines II: Integration, in *Numerische Methoden der Approximationstheorie*, L. Collatz, G. Meinardus, and H. Werner, eds., Birkhäuser Verlag, Basel, 1976, 123–146.
5. Craven, P. and G. Wahba, Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation, Numer. Math. **31** (1979), 377–403.
6. Dierckx, Paul, A fast algorithm for smoothing data on a rectangular grid using spline functions, SIAM J. Numer. Anal **19** (1982), 1286–1304.
7. Dyn, N. and D. Levin, Bell shaped basis functions for surface fitting, in *Approximation Theory and Applications*, Z. Ziegler (ed.), Academic Press, New York, 1981, 113–129.
8. Dyn, N., D. Levin, and S. Rippa, Surface interpolation and smoothing by thin plate splines, in *Approximation Theory IV*, C. Chui, L. Schumaker, and J. Ward (eds.), Academic Press, New York, 1983, 445–449.
9. Dyn, N., D. Levin, and S. Rippa, Numerical procedures for global surface fitting of scattered data by radial functions, SIAM J. Sci. Stat. Comp. **7** (1986), 639–659.
10. Dyn, N. and G. Wahba, On the estimation of functions of several variables from aggregated data, SIAM J. Math. Anal. **13** (1982), 134–152.
11. von Golitschek, M., F. Schardt, and M. Wiegand, Mathematische Auswertung ergospirometrischer Messungen, in *Numerical Methods of Approximation Theory*, L. Collatz, G. Meinardus, and G. Nürnberger (eds.), Birkhäuser, Basel, 1987,

12. Golub, G., M. Heath, and G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, Technometrics **21** (1979), 215–222.
13. Hutchinson, M. and F. DeHoog, Smoothing noisy data with spline functions, Numer. Math. **47** (1985), 99–106.
14. Lyche, T. and L. L. Schumaker, Computation of smoothing and interpolating splines via local bases, SIAM J. Numer. Anal. **10** (1973), 1027–1038.
15. Pope, S. B. and R. Gadh, Fitting noisy data using cross-validated cubic smoothing splines, Tech. Rpt. FDA-87-2, Cornell Univ., 1987.
16. Ragozin, D., Error bounds for derivative estimates based on spline smoothing of exact or noisy data, J. Approx. Th. **37** (1983), 335–355.
17. Reinsch, C. H., Smoothing by spline functions, Numer. Math. **10** (1967), 177–183.
18. Reinsch, C. H., Smoothing by spline functions II., Numer. Math. **16** (1971), 451–454.
19. Rice, J. and M. Rosenblatt, Integrated mean square error of a smoothing spline, J. Approx. Th. **33** (1981), 353–369.
20. Schumaker, L. L., *Spline Functions: Basic Theory*, Wiley Interscience, New York, 1981.
21. Schumaker, L. and F. Utreras, Asymptotic properties of complete smoothing splines and appications, SIAM J. Sci. Stat. Comp. **9** (1988), 24–31.
22. Utreras, F., Natural spline functions: their associated eigenvalue problem, Numer. Math. **42** (1983), 107–117.
23. Utreras, F., On generalized cross validation for multivariate smoothing spline functions, SIAM J. Sci. Stat. Comp.
24. Wahba, G., Smoothing noisy data with spline functions, Numer. Math. **24** (1975), 383–393.
25. Wahba, G., Convergence rates of thin plate smoothing splines when the data are noisy, in *Smoothing Techniques in Curve Estimation*, M. Rosenblatt and T. Glasser (eds.), Lecture Notes 757, Springer-Verlag, Heidelberg, 1979,
26. Wahba, G., Spline bases, regularization and generalized cross-validation for solving approximation problems with large quantities of noisy data, in *Approximation Theory IV*, E. Cheney (ed.), Academic Press, New York, 1980, 905–912.
27. Wahba, G., Ill-posed problems: numerical and statistical methods for mildly, moderately, and severely ill-posed problems with noisy data, in *Proceedings of the International Conference on Ill-posed Problems*, M. Nashed (ed.).
28. Wahba, G., Surface fitting with scattered, noisy data on Euclidean *d*-spaces and on the sphere, Rocky Mt. J. **14** (1984), 281–299.
29. Wahba, G., Cross-validated methods for the estimation of multivariate functions from data on functionals, in *Statistics: An Appraisal*, H. A. David and H. T. David (eds.), Iowa State Univ. Press, 1984, 205–235.
30. Wahba, G. and J. Wendelberger, Some new mathematical methods for variational objective analysis using splines and cross validation, Mon. Wea. Rev. **108** (1980), 36–57.

31. Wahba, G. and S. Wold, A completely automatic French curve: Fitting spline functions by cross-validation, Comm. Stat. 4 (1975), 1–17.

32. Wahba, G. and S. Wold, Periodic splines for spectral density estimation: The use of cross-validation for determining the degree of smoothing, Comm. Stat. 4 (1975), 125–141.

A SEMIINFINITE PROGRAMMING ALGORITHM FOR CONSTRAINED BEST APPROXIMATION

KEN W. BOSWORTH
Department of Mathematics and Statistics
Utah State University

Abstract The general linearly constrained best approximation problem with arbitrary norm is cast as a semiinfinite linear programming problem, using a device of A. Haar (1918) . The resulting optimization problem - that of maximizing a linear form over a convex set described by the intersection of its supporting halfspaces - can be solved numerically by a 2-phase hybrid "interior point / cutting plane" algorithm. A brief outline of both phases of the algorithm is given. Phase 1 is concerned with an "activity" analysis of the constraints posed on the desired fit, and with the construction of an initial feasible solution. Phase 2 produces a sequence of feasible (interior) *and* infeasible (exterior) solutions, both sequences converging to the optimal constrained solution. The method is competitive in *non*-L_2 settings, especially when the norm is not smooth and/or the constraints posed are particularly active.

Key words: Constrained approximations, Semiinfinite programming, Interior point algorithm.

1. Introduction

1.1 Rationale

Researchers in the empirical sciences are faced with the general problem of "fitting" data in a pleasing , rational, and efficient manner. One means of attack is to cast the data fitting problem as a constrained best approximation problem in a normed linear space. The constraints should express various shape qualities and interpolation conditions that the data fit must possess to be "pleasing" or useful. The choice of norm is dictated by both the form of the hypothesized or suspected errors on the data and by the application intended for the fit after it has been produced. The choice of approximating space is made on the basis of one or several of the following criteria: computational ease, model predictions, degree of approximation, and manipulative ease in later applications.

Although algorithms exist for solving constrained best approximation problems for certain specific combinations of constraints, norm, and approximating family, no general all purpose algorithm has been available. This is the motivation for the present work; framing the constrained approximation problem as a semiinfinite convex programming problem posessing special structure, and developing a numerical algorithm that could then be applied in any situation that may obtain.

1.2 The linearly constrained best approximation problem

In this section the data fitting problem is cast in a mathematical framework, flexible enough to handle most practical situations. Let $\{f_1, f_2, \ldots, f_n\}$ be n linearly independent, real valued functions defined and continuous on the compact set $\mathbf{I} \subset R^k$. The linear space $\mathcal{V} := \text{lin}\{f_1, f_2, \ldots, f_n\}$ will be referred to as the approximating subspace, and the set $\{f_1, f_2, \ldots, f_n\}$ as the approximating family. Let f denote real data, defined and continuous on the set $\mathbf{I}$; i.e., $f \in C[\mathbf{I}]$. If one wishes to fit the data f using elements from $\mathcal{V}$, one needs a criterion for measuring the goodness of the fit; this measure is usually a norm or seminorm defined on the linear space $\mathcal{W}$ of functions gotten by adjoining f to $\mathcal{V}$; and is denoted by $||h||$, where $h \in \mathcal{W}$. The problem of finding $g_{ba} \in \mathcal{V}$ such that:

$$||f - g_{ba}|| = \inf_{g \in \mathcal{V}} ||f - g||$$

is the classical problem of best approximation from a linear subspace. The element g_{ba} is called the best "$||\cdot||$ - approximation" to f from $\mathcal{V}$. Using standard compactness and continuity arguments, one can easily prove the existence of such an element $g_{ba} \in \mathcal{V}$ for each $f \in C[\mathbf{I}]$.

However, one often desires that the best fit to the data also possess other shape or structural qualities. In a large majority of cases, such constraints can be formulated in terms of an indexed collection of linear constraints of the form:

$$a(p) \leq L(p; f, g) \leq b(p), \forall p \in \mathbf{J}$$

where:

$$a(p) : \mathbf{J} \rightarrow R^e,$$
$$b(p) : \mathbf{J} \rightarrow R^e,$$

with R^e the extended reals; $L(p; f, g)$ is a continuous linear functional acting on f or $g \in \mathcal{V}$ or both, indexed by and continuous in $p \in \mathbf{J}$; and $\mathbf{J}$ is a compact topological space, called the *index set*, often identical with $\mathbf{I}$ or a subset of $\mathbf{I}$. The function a (resp. b) is required to be u.s.c. (resp. l.s.c.) on $\mathbf{J}$, continuous on $\mathbf{J} - a^{-1}\{-\infty\}$ (resp. $\mathbf{J} - b^{-1}\{\infty\}$), and to avoid vacuous problems, $a(p) \leq b(p)$, $\forall p \in \mathbf{J}$.

In the case where $\mathbf{I} \subset R^1$, one has as common examples of such linear constraints (assuming $g \in \mathcal{V}$ and f are sufficiently smooth):

1. **Interpolation Conditions:** the approximant g is to exactly agree with the data f at the specified points $p \in \mathbf{J} \subseteq \mathbf{I}$. Then take $a(p) = b(p) = 0$, and

$$L(p; f, g) := f(p) - g(p), \ \forall p \in \mathbf{J}.$$

2. **Non-negativity Constraints:** the approximant g is to remain non-negative over some subset $\mathbf{J} \subseteq \mathbf{I}$. Set $a(p) = 0$, $b(p) = \infty$, and

$$L(p; f, g) := g(p), \ \forall p \in \mathbf{J}.$$

3. **Specified Moments:** the approximant g is to have its p^{th} moments, $p \in \mathbf{J} \subset Z$, specified as either the same as that of the data, f, or as some value m_p . In the first instance, take $a(p) = b(p) = 0$ for $p \in \mathbf{J}$, and

$$L(p; f, g) := \int_{\mathbf{I}} x^p(f(x) - g(x))dx.$$

In the second instance, take $a(p) = b(p) = m_p$ for $p \in \mathbf{J}$, and

$$L(p; f, g) := \int_{\mathbf{I}} x^p g(x)dx.$$

Notice that in examples 1 and 2, it happened that $\mathbf{J} \subseteq \mathbf{I}$, whereas example 3 shows that $\mathbf{J}$ may just as well be an arbitrary index set. One is free to impose several of the above conditions simultaneously in any approximation problem, and then the index set $\mathbf{J}$ might contain several "copies" of $\mathbf{I}$ or subsets of $\mathbf{I}$.

1.3 Haar's transcription

To transcribe the constrained best approximation problem:

$$\mathcal{P}_c : \begin{cases} \min_{g \in \mathcal{V}} \ \|f - g\| \\ a(p) \leq L(p; f, g) \leq b(p), \ \forall p \in \mathbf{J} \end{cases}$$

into a semiinfinite linear program, one introduces homogeneous coordinates $\vec{x} := (x_1, \ldots, x_n, x_{n+1})$, $x_{n+1} > 0$, as follows. For each $g \in \mathcal{V}$, $g = \sum_{i=1}^n \alpha_i f_i$, and $f - g = f - \sum_{i=1}^n \alpha_i f_i$, so $e := f - g \in \mathcal{W}$. In $\mathcal{W}$, consider now all linear combinations of the form:

$$w = \sum_{i=1}^n x_i f_i \ + x_{n+1} f, \qquad x_{n+1} > 0,$$

which we write as:

$$w = \vec{x} \cdot \vec{f}, \qquad \vec{e}_{n+1} \cdot \vec{x} > 0$$

where:

$$\begin{aligned} \vec{x} &:= (x_1, \ldots, x_n, x_{n+1}), \\ \vec{f} &:= (f_1, \ldots, f_n, f), \\ \vec{e}_{n+1} &:= (0, \ldots, 0, 1), \end{aligned}$$

and the dot product is the standard scalar product on R^{n+1} . Homogeneous is a suitable adjective for the $\vec{x}$ coordinates, as each ray $\vec{r}_{\vec{u}} := \{\vec{x} \mid \vec{x} = t\vec{u},\ t > 0\} \subset H$, where $H := \{\vec{x} \mid \vec{e}_{n+1} \cdot \vec{x} > 0\}$, corresponds to precisely one error function $e := f - g$, $g = \sum_{i=1}^{n} \alpha_i f_i$, via the well defined map: $\alpha_i = -x_i/x_{n+1} = -u_i/u_{n+1}$, $i = 1, \ldots, n$.

Now, defining $\beta_i(p) := L(p; 0, f_i)$ for $i = 1, \ldots, n$ and $\beta_{n+1}(p) := L(p; f, 0)$, the collection of constraints in $\mathcal{P}_c$ becomes, in homogeneous coordinates:

$$a(p)x_{n+1} \le -\sum_{i=1}^{n} \beta_i(p)x_i \ + \beta_{n+1}(p)x_{n+1} \le b(p)x_{n+1}, \qquad \forall p \in \mathrm{J}.$$

Defining the vectors:

$$\begin{gathered} \vec{n}_+(p) := (\beta_1(p), \ldots, \beta_n(p), \beta_{n+1}(p) - a(p)), \\ \text{for all } p \in \mathrm{J} - a^{-1}\{-\infty\}, \\ \vec{n}_-(p) := (-\beta_1(p), \ldots, -\beta_n(p), -\beta_{n+1}(p) + b(p)), \\ \text{for all } p \in \mathrm{J} - b^{-1}\{\infty\}, \end{gathered}$$

with $\vec{n}_+(p) := \vec{0}_{n+1}$ on $a^{-1}\{-\infty\}$ and $\vec{n}_-(p) := \vec{0}_{n+1}$ on $b^{-1}\{\infty\}$, the homogeneous variable transcription of the constraints is the closed convex cone:

$$C := \{\vec{x} \in R^{n+1} \mid \vec{n}_+(p) \cdot \vec{x} \le 0,\ \vec{n}_-(p) \cdot \vec{x} \le 0,\ \forall p \in \mathrm{J}\}.$$

That is to say, any vector $\vec{x} \in C \cap H$ corresponds under the above rule of correspondence to a function $g \in \mathcal{V}$ satisfying the totality of constraints imposed on $\mathcal{P}_c$. Moreover, any $g \in \mathcal{V}$ *not* satisfying some constraint has homogeneous coordinates *not* in $C \cap H$.

Lastly, define the closed convex body (called by Haar the "Eichkörper"):

$$K := \{\vec{x} \in R^{n+1} \mid \|\vec{x} \cdot \vec{f}\| \le 1\},$$

and consider the semiinfinite linear program:

$$\mathcal{P}_h : \begin{cases} \max\ \vec{e}_{n+1} \cdot \vec{x} \\ \vec{x} \in K \cap C \end{cases}$$

With the above notation and definitions, one has the simple result:

Result 1. Problem $\mathcal{P}_c$ and $\mathcal{P}_h$ are equivalent in the following sense:

1. $\mathcal{P}_c$ is inconsistent if and only if $\mathcal{P}_h$ has optimal solution $\vec{x}_{opt} = \vec{0}_{n+1}$.

2. $\mathcal{P}_c$ is consistent and has $\|f - g_{opt}\| = 0$ if and only if $\mathcal{P}_h$ has unbounded feasible solutions and unbounded objective values; that is, there exists a ray $r_{\vec{u}} := \{\vec{x} = t\vec{u},\ t \geq 0, \vec{e}_{n+1} \cdot \vec{u} > 0\} \subset K \cap C$. In this case, any g_{opt} corresponds to some such $\vec{u}$ under the map $g = \sum_{i=1}^{n} \alpha_i f_i$ with $\alpha_i := -u_i/u_{n+1}$ and vice versa: any such ray $r_{\vec{u}}$ yields an optimal solution g_{opt} by the same map.
3. $\mathcal{P}_c$ is consistent and has $\|f - g_{opt}\| = E > 0$ if and only if $\mathcal{P}_h$ has a finite optimal solution $\vec{x}_{opt} = \vec{v}$ with $\vec{e}_{n+1} \cdot \vec{v} = v_{n+1} > 0$. In this case, the error in the constrained best approximation is $E = 1/v_{n+1}$. Any constrained best approximation g_{opt} corresponds to some optimal vector $\vec{v}$ under the map $g = \sum_{i=1}^{n} \alpha_i f_i$ with $\alpha_i := -v_i/v_{n+1}$ and vice versa: any such optimal vector $\vec{v}$ yields an optimal solution g_{opt} by the same map.

The proof of this result is straightforward, and is in fact a direct application of a minimum norm duality theorem of Deutsch and Maserick (1967) and found in Luenberger (1969), page 119, and thus will not be given here.

2. A 2 phase algorithm

For convenience in what follows, the supporting normal vectors appearing in C are renamed and reindexed, omitting any that are $\vec{0}_{n+1}$, i.e., $C = \{\vec{x} \in R^{n+1} \mid \vec{n}(p) \cdot \vec{x} \leq 0,\ p \in \mathbf{J}'\}$. Also, set $C \cap \overline{H} := C_H$.

2.1 Phase 1

Problem $\mathcal{P}_h$ can be efficiently solved by an interior point algorithm to be described in the following section *provided* $K \cap C_H$ has interior in R^{n+1}. As K always has interior, int$(K \cap H)$ is empty iff int$(C_H) = \emptyset$. In terms of the constraints in $\mathcal{P}_c$, this situation obtains when either interpolatory conditions are imposed (not satisfied by *all* $g \in \mathcal{V}$), *or* when a collection of "one-sided" constraints has "ganged up" to form a " generalized" interpolation condition or conditions. As the solution of general problems is the goal, Phase 2 is preceded by an analysis of the cone C_H determining the dimension of its relative interior, followed, if necessary, by a suitable introduction of a reduced set of basis functions $\{\tilde{g}_1, \ldots, \tilde{g}_{\tilde{n}}\}, \tilde{n} < n$, with linear span $\tilde{\mathcal{V}} \subset C$, such that *all* interpolation conditions are automatically satisfied by any $\tilde{g} \in \tilde{\mathcal{V}}$. Hence, by adopting $\tilde{\mathcal{V}}$ as the approximating space, the resulting cone $\tilde{C}_H$ will have interior in the reduced dimension space $R^{\tilde{n}+1}$.

It turns out that the key to Phase 1 is determining whether $\vec{0}_{n+1}$ is in the convex hull of $\vec{e}_{n+1} \cup \{\vec{n}(p)\}_{p \in \mathbf{J}'}$. If so, one determines the set L of all such vectors which are involved in nontrivial convex combinations yielding $\vec{0}_{n+1}$. The resulting set L has orthogonal compliment $L^{\perp}$ containing C_H. In essence, the set L corresponds precisely to the constraints (if any) which have "ganged up" in $\mathcal{P}_c$ to reduce the degrees of freedom in the approximating space. That is, a basis for $L^{\perp}$ yields, under the inverse of the homogeneous coordinate transcription, a reduced basis $\{\tilde{g}_1, \ldots, \tilde{g}_{\tilde{n}}\}$ to work in, with $\dim(L^{\perp}) = \tilde{n}$, or shows, in case $\tilde{n} = 0$, that the problem $\mathcal{P}_c$ is overconstrained. Fortunately, such a task is efficiently undertaken; see Bosworth (1988) for a detailed algorithm description. Moreover, if L is empty, then the just cited algorithm provides one with a vector $\vec{u} \in C_H$, separating $\vec{0}_{n+1}$ from the convex hull of $\vec{e}_{n+1} \cup \{\vec{n}(p)\}_{p \in J'}$.

2.2 Phase 2: the tunnelling algorithm

A description is now given of the upper level structure of the actual Phase 2 semi-infinite programming algorithm solving the problem $\mathcal{P}_h$. A detailed description of Phase 2 can be found in Bosworth (1984) . Note that on successful completion of Phase 1, a vector $\vec{u}$ pointing into the interior of $K \cap C_H$ is available, and that $\vec{0}_{n+1} \in K \cap C$. Note also that in the course of Phase 1, the dimension n of the problem has been suitably redefined if necessary. Initialize $\vec{x}_{cm} := \vec{0}_{n+1}$.

2.2.1 Shooting

From the point $\vec{x}_{cm}$, "shoot" in the positive $\vec{u}$ direction to $s\vec{x} \in \partial(K \cap C)$. If no such point exists, then $\vec{u}$ satisfies Result 1, condition 2, and the algorithm terminates. If $s\vec{x} \notin \partial K$, then scale $s\vec{x}$ positively until it reaches ∂K. Again, if no such point exists, then the direction vector $s\vec{x}$ satisfies Result 1, condition 2, and the algorithm terminates.

At $s\vec{x}_1 := s\vec{x}$ compute a unit outer supporting normal $s\vec{n}_1$ to ∂K, using the subgradient $\partial\|\vec{x} \cdot \vec{f}\|$ at $s\vec{x}_1$. If more than one normal exists (nonsmooth norm), select one with maximal $n + 1^{st}$ component. If the normal $s\vec{n}_1 = \vec{e}_{n+1}$, then $s\vec{x}_1$ is the optimal solution to $\mathcal{P}_h$, Result 1, case 3, and the algorithm terminates.

2.2.2 Webbing

Set $M := \{\vec{x} \in R^{n+1} | \, \vec{e}_{n+1} \cdot \vec{x} = \vec{e}_{n+1} \cdot s\vec{x}_1\}$ and $N := \{\vec{x} \in R^{n+1} | \, \vec{e}_{n+1} \cdot \vec{x} = 0\}$. I.e., $M = N + s\vec{x}_1$.

A collection of $n+1$ supporting halfspaces to the set $K \cap C$ with support points on $\partial(K \cap C) \cap M$ is constructed by means of a n stage recursion (using the shooting technique of the previous subsection), the aim being to "equidistribute" the support points on $\partial(K \cap C)$. These halfspaces will be used in the following stage of the algorithm to define a *minimal* linear program approximating $\mathcal{P}_h$.

The k^{th} stage of the recursion ($k = 2, \ldots, n+1$) is "roughly" the following:

$$\begin{aligned} \vec{x}_{cm,k-1} &:= \tfrac{1}{k-1} \textstyle\sum_{i=1}^{k-1} s\vec{x}_i \\ \vec{n}_i &:= \operatorname{proj}_N s\vec{n}_i \, , \; i = 1, \ldots, k-1, \end{aligned}$$

define the k^{th} search direction $\vec{u}_k$ by:

$$\begin{aligned} \vec{u}_k \cdot \vec{n}_i &= -1 \, , \; i = 1, \ldots, k-1 \\ \vec{u}_k &\in \operatorname{lin}\{\vec{n}_1, \ldots, \vec{n}_{k-1}\}, \end{aligned}$$

and then the supporting vector $s\vec{x}_k$ and outer supporting normal $s\vec{n}_k$ to $K \cap C$ are found by "shooting" from $\vec{x}_{cm,k-1}$ in the positive $\vec{u}_k$ direction. However, here

$s\vec{n}_i$ is allowed to be a supporting normal to the cone C. (If the vectors $\{\vec{n}_i\}_{i=1}^{k-1}$ are dependent, randomly chosen vectors in N are added to the set until a well defined direction $\vec{u}_k$ can be obtained from the above relations, hence the adjective "roughly".)

By construction, $K \cap C \subseteq \{\vec{x} \in R^{n+1} | \; s\vec{n}_i \cdot \vec{x} \leq s\vec{x}_i \cdot s\vec{x}_i\}$ for $i = 1, \cdots, k$. At the conclusion of the recursion, set $\vec{x}_{cm} := \frac{1}{n+1} \sum_{i=1}^{n+1} s\vec{x}_i$.

2.2.3 Analysis of a minimal linear program

Consider the simple linear program:

$$\mathcal{P}_{mlp} : \left\{ \begin{array}{lll} \max & \vec{e}_{n+1} \cdot \vec{x} & \\ s\vec{n}_i \cdot \vec{x} & \leq s\vec{n}_i \cdot s\vec{x}_i & i = 1, \ldots, n+1 \end{array} \right.$$

By construction, $K \cap C$ is within the feasible region defined by $\mathcal{P}_{mlp}$, and hence an optimal value of $\mathcal{P}_{mlp}$ provides an upper bound on the optimal value of $\mathcal{P}_h$. Moreover, $\mathcal{P}_{mlp}$ is trivial to analyze (one "Simplex Pivot"). Since $\mathcal{P}_{mlp}$ can be considered to be a "coarse" discretization of the semiinfinite program $\mathcal{P}_h$, it will be reasonable to use any information obtained from the optimality analysis of $\mathcal{P}_{mlp}$ as approximating the situation in $\mathcal{P}_h$.

Four cases may obtain in $\mathcal{P}_{mlp}$:

1. There exists a unique optimal solution $\vec{v}_{mlp}$. Then the feasible region in $\mathcal{P}_{mlp}$ intersected with $\overline{H}$ is a bounded polytope containing $K \cap C_H$. The solution $\vec{v}_{mlp}$ can be considered an approximate solution to $\mathcal{P}_h$. Define a new search direction for the "shooting" stage, $\vec{u} := \vec{v}_{mlp} - \vec{x}_{cm}$.
2. There exists infinitely many optimal solutions $\vec{w}_{mlp}$, all with the same $n+1^{st}$ component. Define a new search direction $\vec{u} := \vec{v}_{mlp} - \vec{x}_{cm}$, where $\vec{v}_{mlp}$ is the closest optimal solution to $\vec{x}_{cm}$. It can be easily shown that this selection of $\vec{u}$ has nonnegative $n + 1^{st}$ component.
3. The system of equalities $A\vec{x} = \vec{b}$ with $\text{row}_i A := s\vec{n}_i$, $b_i := s\vec{n}_i \cdot s\vec{x}_i$, $i = 1, \ldots, n+1$ is singular, but has vectors $\vec{w} \in \text{Kern}(A)$ with positive $n + 1^{st}$ component. Select the unit vector $\vec{u} \in \text{Kern}(A)$ with maximal $n + 1^{st}$ component as the new shooting direction.
4. The system $A\vec{x} = \vec{b}$ is nonsingular, but $\mathcal{P}_{mlp}$ has unbounded objective values. Then it can be shown that there exists a ray $r_{\vec{u}}$ of unbounded feasible solutions to $\mathcal{P}_{mlp}$ with unit direction vector $\vec{u}$ having maximal positive $n+1^{st}$ component. This is the new shooting direction.

In each of the 4 cases, a vector $\vec{u}$ is obtained, which heuristically points from $\vec{x}_{cm} \in K \cap C$ in a direction of "ascent" for $\mathcal{P}_h$.

2.2.4 Convergence

If, in case 1 or 2 above, $\vec{v}_{mlp} - \vec{x}_{cm}$ is "sufficiently small", then one can conclude convergence. If $\vec{e}_{n+1} \cdot \vec{x}_{cm} < (1/\text{tol})$, where tol is an upper bound on the error needed in a near-best approximation, then one concludes convergence. In either case, to obtain the final $\vec{v}$ to use in computing g_{ba}, shoot one last time from $\vec{x}_{cm}$ in the $\vec{u}$ direction to $\vec{v} \in \partial(K \cap C)$.

Else, one returns to the shooting stage, and continues with the algorithm.

3. Conclusions

Linear convergence of the algorithm can be concluded provided Phase 1 is successfully completed; see Bosworth (1984). That is, the sequence of points $\{\vec{x}_{cm}\}$ converges in objective value, from below, to the optimal value for $\mathcal{P}_h$ at least at a linear rate. If $\mathcal{P}_h$ is in situation 3 of Result 1, then one is also guaranteed the existence of subsequences of $\{\vec{x}_{cm}\}$ and $\{\vec{v}_{mlp}\}$ having convergence in R^{n+1} to an optimal solution $\vec{v}$ of $\mathcal{P}_h$ (different subsequences can have different limits; however, all tend to the same objective value). In polyhedral norm situations, with a finite set of constraints, the algorithm converges in a finite number of steps, often outperforming the Dual Simplex Algorithm of linear programming, due to its ability to bypass in one step several pivots taken by the Dual Simplex Algorithm. (The Dual Simplex Algorithm computes a sequence of infeasible solutions, converging to the optimal solution of $\mathcal{P}_h$, similar to the solutions $\{\vec{v}_{mlp}\}$ computed here.)

Numerical results for the case of uniform norm shape constrained approximation problems have appeared in Bosworth (1987) . Applications to the l_1 and other specially tailored norms will appear shortly. One final observation is that the viability of the algorithm appears most strongly in *non* l_2 settings. In constrained l_2 settings, the code of Lawson and Hanson (1974) is to be recommended.

References

1. Ken W. Bosworth. *A general method for the computation of best uniform norm approximations*. PhD thesis, Rensselaer Polytechnic Institute, Troy, N.Y., 1984.
2. Ken W. Bosworth. *A numerical algorithm for the determination of the linear span of a cone given in polar form*. Technical Report, Utah State Univ. , Dept. of Mathematics and Statistics, 1988.
3. Ken W. Bosworth. Shape constrained curve and surface fitting. In G. E. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 247-263, SIAM, 1987.
4. F.R. Deutsch and P.H. Maserick. Applications of the Hahn-Banach theorem in approximation theory. *SIAM Review*, 9(3):516-530, July 1967.
5. A. Haar. Über die Minkowskische Geometrie und die Annäherung an stetige Funktionen. *Math. Ann.*, 78:294-311, 1918.
6. C. L. Lawson and R. J. Hanson. *Solving Least Square Problems*. Prentice Hall, 1974.
7. D.L. Luenberger. *Optimization by Vector Space Methods*. John Wiley, 1969.

INFERENCE REGION FOR A METHOD OF LOCAL APPROXIMATION BY USING THE RESIDUALS

M. BOZZINI
Dipartimento di Matematica
Universita' di Lecce
L. LENARDUZZI
Istituto Applicazioni Matematica ed Informatica C. N. R.

Abstract We consider a formula for smoothing noisy data in which the variances are different. The formula is obtained by a local weighted approximation. The weight is a function of a smoothing parameter α, which varies with the point. Then a diagnostic band for data analysis is obtained : this determines whether an initial and computationally quick choice of α constant for all the data allows a homogeneous accuracy of approximation on the whole domain. In the case when it is necessary for α to be variable, a fast algorithm to evaluate α is proposed.

Key words: Inference region, Weighted local approximation, Smoothing, Variable smoothing parameter, Data analysis.

1. Introduction

In the problem of approximating a function from a sample of function values, it is of interest to know the reliability achieved with the chosen approximation method. In other words what is the probability that the approximating function fits the unknown function to a specified accuracy?

On this subject we refer to the work by Wahba (1983), who, on the basis of a Bayesian model (that is to say the region is that within which the approximating function can fall with that sampling) for spline functions, assuming that the noise variance is constant determines an inference region with constant band. A later work by Silverman (1985) generalizes the work by Wahba by considering more general data configurations and noise characterized by variance $\sigma^2 = \sigma^2(x)$ as a function of the abscissa x. The band obtained varies according to the data location density and variance.

Müller and Stadtmüller (1987) study a method with a localized smoothing parameter based on estimates of the higher derivatives of $f(x)$.

In this note, on the basis of an approximation method with constant window, we construct an inference region for the unknown function $f(x)$ (in other words $f(x)$ falls in the region derived from that sampling with given probability) by using samples of random variables $\tilde{f}_r(x)$ $r = 1, \ldots, l$ $l \geq 1$ and finite, with $E(\tilde{f}_r(x)) = f(x)$ and covariance matrix $\sigma_{rs}^2 = \delta_{rs}\sigma_r^2$.

The method, which is computationally simple, has variable band and indicates the regions where more accuracy is needed when constructing the approximating function.

Then a computationally straightforward method is suggested to improve the approximation locally, by modifying the local smoothing parameter.

2. Approximation formula

Let a set $J = \{x_i, \tilde{f}(x_i)\}_1^N$ be given on a domain $D = [a, b] \subset R$. An approximation to $f(x)$ is required.

We assume that the function values $\tilde{f}(x_i)$ are the result of sampling from l populations with sample sizes $n_1, n_2, \ldots, n_l$, respectively, each one according to a uniform distribution (for simplicity) and $\sum_1^l n_r = N$ and such that

$$E(\tilde{f}_r(x)) = f(x),$$
$$\sigma_{rs}^2 = \delta_{rs}\sigma_r^2.$$

For simplicity, in the following it is assumed that $n_r = n$ for $r = 1, \ldots, l$.

Let us consider the method studied in Bozzini and Lenarduzzi (1988a) to approximate the unknown function $f(x)$.

Precisely, after indicating with x_i a generic assigned point , take a neighbourhood $I = [x_i - R, x_i + R]$ and determine the constant function $\tilde{c}_i$ by the weighted least squares method:

$$min \sum_{x_j \in I} w_{j\alpha}(c - \tilde{f}(x_j))^2$$

where the weights $w_{j\alpha}$ have the following expression:

$$w_{j\alpha} = \begin{cases} \frac{1}{(d^2+\alpha)\sigma_r^2} & if\ d < R, \\ 0 & if\ d \geq R, \end{cases}$$

with $d^2 = dist^2(x_i, x_j)$ the Euclidean distance and α a positive smoothing parameter.

Then the approximating function

$$\tilde{m}(x) = \frac{\sum_1^N \tilde{c}_i \phi_i(x)}{\sum_1^N \phi_i(x)} \tag{1}$$

is constructed.

Here the $\phi_i(x)$ are smooth functions with support $(x_i - R_1, x_i + R_1)$ with $R_1 < R$ and such that $\tilde{m}(x_i) = \tilde{c}_i$.

This approximating function depends on α; in particular, as $\alpha \to 0$ the function $\tilde{m}(x)$ tends to an interpolating function, while, as $\alpha \to \infty$ $\tilde{m}(x)$ tends to the function in which the constants $\tilde{c}_i$ are solution of the weighted least squares problem with weights $w_{ir} = \frac{1}{\sigma_r^2}$.

3. Inference region

Let us assume we work in asymptotic conditions (that is to say with $N \to \infty, R \to 0, NR \to \infty$) and that we also have information beyond the boundary of D.

After constructing the function $\tilde{m}(x)$ on the basis of (1) we determine a computable inference region.

From the Markov inequality (see Ross (1980)), one gets for $i = 1, \ldots, N$:

$$Prob\{\mid f(x_i) - \tilde{m}(x_i) \mid < k\{E_e[f(x_i) - \tilde{m}(x_i)]^2\}^{\frac{1}{2}}\} > 1 - \frac{1}{k^2}.$$

The expected mean squared error $E_e[f(x_i) - \tilde{m}\ (x_i)]^2$, which will be called $EMSE_i$ in what follows, has the expression:

$$EMSE_i = \{E_e[f(x_i) - \tilde{m}(x_i)]\}^2 + var(\tilde{m}(x_i))$$

and, of course, it is a function of the parameter α.

In Bozzini and Lenarduzzi (1988a) it was proved that there is one and only one value α_i^* of α that minimizes $EMSE_i(\alpha)$:

$$EMSE_i(\alpha_i^*) = min_{\alpha>0} EMSE_i(\alpha)$$

and such a value falls in$(0, R^2)$ or in $[R^2, \infty)$ depending on whether the ratio

$$\frac{R^4 f_i^{(2)2}}{\left[\frac{\delta}{2nR}\right]}$$

is greater or less than one, (here $\delta = (\sum' \frac{1}{\sigma_r^2})^{-1}$).

Furthermore, using the hypothesis that $f(x)$ can be developed as a power series, the principal part of $EMSE_i(\alpha_i^*)$ has the following expression:

$$EMSE_i(\alpha_i^*\ ) \simeq \frac{f_i^{(2)2}}{\pi^2} R^2 \alpha_i^* + \frac{\delta}{2n\sqrt{\alpha_i^*}\pi}$$

if $\alpha_i^* \in (0, R^2)$ and

$$EMSE_i(\alpha_i^*) \simeq \frac{f_i^{(2)2} R^4(\frac{1}{36} - \frac{R^2}{30\alpha_i^*}) + \frac{\delta}{2nR}(1 - \frac{2R^2}{3\alpha_i^*})}{(1 - \frac{R^2}{3\alpha_i^*})^2}$$

if $\alpha_i^* \in [R^2, \infty)$

It can be verified analytically that an upper bound to the principal part of $EMSE_i(\alpha_i^*)$ is:

$$1.5\delta p_{ii}(\alpha_i^*), \tag{2}$$

where $p_{ii}(\alpha)$ is the asymptotic value of the weight $w_{i\alpha}/\sum w_{j\alpha}$ (index of influence of the i-th datum) divided by $\frac{\delta}{\sigma_i^2}$ and associated to the value $\tilde{c}_i$ at the point x_i.

Consider the random variable

$$\theta_{\alpha_i} = \frac{\sum^{N_I} \beta_i^2(\tilde{f}_i - \tilde{m}_{\alpha_i}(x_i))^2}{\sum^{N_I}(1 - p_{ii}(\alpha_i))} \tag{3}$$

with $\beta_i^2 = \frac{\delta}{\sigma_i^2}$;

then:

$$E_e(\theta) = \delta + \frac{\sum^{N_I} \beta_i^2 EMSE_i}{\sum^{N_I}(1 - p_{ii})} - \delta\frac{\sum^{N_I} p_{ii}}{\sum^{N_I}(1 - p_{ii})}.$$

If one assumes that the second derivative does not change too much on I then

$$AMSE_I(\alpha_i) = E_e \int_I (f(t) - \tilde{m}(t))^2 dt = 2R\, EMSE_i(\alpha_i) + O(R^6)$$

In optimal conditions (that is to say $R \asymp n^{-\frac{1}{5}}, \alpha_i^* \asymp R^2$), it has been proved in Bozzini and Lenarduzzi (1988a)that $EMSE_i(\alpha_i^*) = O(n^{-\frac{4}{5}})$ and $p_{ii} = O(n^{-\frac{4}{5}})$; so:

$$E_e(\theta(\alpha_i^*)) = \delta + O(n^{-\frac{4}{5}}). \tag{4}$$

As can be seen by developing the calculations, one has:

$$var(\theta(\alpha_i^*)) = O(n^{-\frac{4}{5}})$$

so that $\theta(\alpha_i^*)$ is consistent estimator for δ.

From (4) it follows that the principal part of $1.5p_{ii}(\alpha_i^*)E_e(\theta(\alpha_i^*))$ is given by $1.5\delta v_{ii}(\alpha_i^*)$. On the basis of (2) and (3) it follows that an inference region which is computable for $f(x)$ at the probability level $(1 - \frac{1}{k^2})$ is given by:

$$\tilde{m}_{\alpha_i^*}(x) - k(1.5\theta_{\alpha_i^*} p_{ii}(\alpha_i^*))^{\frac{1}{2}} \le f(x) \le \tilde{m}_{\alpha_i^*}(x) + k(1.5\theta_{\alpha_i^*} p_{ii}(\alpha_i^*))^{\frac{1}{2}}. \tag{5}$$

In order to make the method more efficient, in the case when the second derivative does not change too much on the domain, it is preferable to calculate a constant value of α for all $x_i \in J$.

This is possible by calculating the value α_{opt} which minimizes $IMSE = E_e \int_D (f(t) - \tilde{m}(t))^2 dt$ (see Bozzini and Lenarduzzi (1988a)).

In this case (5) becomes:

$$\tilde{m}_{\alpha_{opt}}(x) - k(1.5\theta_{\alpha_{opt}} p_{ii}(\alpha_{opt}))^{\frac{1}{2}} \leq f(x) \leq \tilde{m}_{\alpha_{opt}}(x) + k(1.5\theta_{\alpha_{opt}} p_{ii}(\alpha_{opt})))^{\frac{1}{2}}. \tag{6}$$

In the case in which, for some x_i, $(f_i^{(2)})^2 \gg (f_M^{(2)})^2 = \int_a^b \frac{f^{(2)2}}{b-a}(t)dt$ then, as derived in Bozzini and Lenarduzzi (1988a), both $EMSE_i(\alpha_i^*)$ and $E(\theta(\alpha_{opt}))p_{ii}(\alpha_{opt})$ increase because of the bias; therefore this band is wider than that for the other points.

Remark:In the case when there is only one variance, that is to say $\sigma_r^2 = \sigma^2$, $r = 1, \ldots, l$, the above remains true provided that one substitutes $w_{j\alpha} = \frac{1}{(d^2+\alpha)\sigma_r^2}$ with $w_{j\alpha} = \frac{1}{d^2+\alpha}$ and δ with σ^2.

4. Variable smoothing parameter

Let us consider a smooth function $f(x)$ which has peaks at some points. Such a function therefore has a second derivative $f_M^{(2)2}$ which is almost constant and given by $\int_a^b \frac{f^{(2)2}}{b-a}dx$ but at the point x_i, $f_i^{(2)2} \gg f_M^{(2)2}$.

The parameter α,which is optimal on the whole interval $[a, b]$, falls in $[R^2, \infty)$ as the method constructs an approximation function in the least squares sense which removes the noise and provides a smooth function. On the contrary, the optimal value of α at the point x_i falls in the interval $(0, R^2)$ in order to give a function which is locally nearer the interpolating function.

Consider the random variable:

$$Z = \frac{\sum_1^N \beta_i^2(\tilde{f}_i - \tilde{m}_i)^2}{\sum_1^N (1 - p_{ii})}.$$

Its expected value is

$$E(Z) = \delta + \frac{IMSE}{\sum(1 - p_{ii})} - \delta\frac{\sum p_{ii}}{\sum(1 - p_{ii})}.$$

In the case we are studying, the optimal value of α for $IMSE$ is α_{opt} and therefore:

$$E(Z_{opt}) = \delta(1 + k_1 n^{-\frac{4}{5}}).$$

Assuming $\alpha_i^* \asymp n^{-\frac{2}{5-\epsilon}}$ one gets:

$$E(\theta(\alpha_i^*)) = \delta(1 + k_2 n^{-\frac{4-\epsilon}{5-\epsilon}}),$$

from which

$$Ratio = \frac{E(Z_{opt})}{E_e(\theta(\alpha_i^* \,))} \simeq 1 - k_2 n^{-\frac{4-\epsilon}{5-\epsilon}}.$$

On the contrary, when one constructs locally the function $\tilde{m}\,(x)$ using the value α_{opt} instead of α_i^* :

$$E_e(\theta(\alpha_{opt}\,)) \simeq \delta(1 + k_3 n^{-\frac{20-7\epsilon}{5(5-\epsilon)}}),$$

from which

$$Ratio = \frac{E(Z_{opt})}{E_e(\theta(\alpha_{opt}\,))} \simeq 1 - k_3 n^{-\frac{20-7\epsilon}{5(5-\epsilon)}}$$

(here the k_j are positive constants).

It is now evident that the band, calculated by using $\theta(\alpha_{opt}\,)$, is no longer an inference band but only a tool for data analysis : in fact on one hand $E(\theta(\alpha_i^*\,))p_{ii}(\alpha_i^*\,)$ and $EMSE_i(\alpha_i^*\,)$ are of the order of $n^{-\frac{4-\epsilon}{5-\epsilon}}$, but on the other $E(\theta(\alpha_{opt}\,))p_{ii}(\alpha_{opt}\,)$ is of the order of $n^{-\frac{4}{5}}$.

In order to obtain a better estimate of the function on that interval and more reliability it is appropriate to modify locally the value of α.

This can be done according to the following law:

$$\alpha = \alpha_{opt} \; * g\{\frac{Z_{opt}}{\theta(\alpha_{opt}\,)}\},$$

where g is an increasing function of its argument and $g(1) = 1$.

5. Numerical results

We refer to two cases from the references.

For the estimates of α_{opt} and σ_r^2 see Bozzini and Lenarduzzi (1988b) .

Case 1: test function from Müller and Stadtmüller (1987):

$$g_2(x) = h(0.25, 0.05) + h(0.5, 0.1),$$

where h is a Gaussian density $N = 100$, $k = 4.48$, $R = .039$, $\sigma_1 = 0.01$, $\sigma_2 = 0.02$.

The function is smooth, so a good approximation accuracy is reached and the confidence band is uniform enough, by using (6). The results are presented in figure 1.

Case 2: test function number 4 in Wahba (1983):

$$f(x) = \begin{cases} 0, & 0 \le x \le \frac{1}{3}, \\ 36(t - \frac{1}{3}), & \frac{1}{3} \le t \le \frac{1}{2}, \\ 36(\frac{2}{3} - t), & \frac{1}{2} \le t \le \frac{2}{3}, \\ 0, & \frac{2}{3} \le t \le 1, \end{cases}$$

normal noise $\sigma = 0.02$, $N = 100$, $k = 4.48$, $R = .049$, $\alpha = 0.01$.

The function presents discontinuity points in the first derivative.

When using (6) to construct the inference band, one observes an enlargement of the band near the discontinuity points (see figure 2).

In figure 3 can be seen the approximation with the global α and the data.

To obtain better accuracy it is possible to use the method described in section 4, with the following function:

$$g(ratio) = (ratio)^{10}.$$

The resulting band is presented in figure 4.

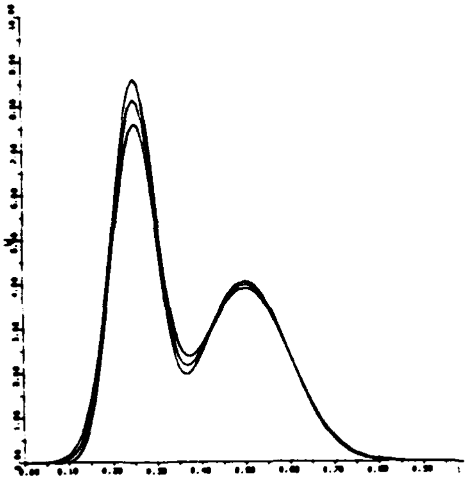


Figure 1

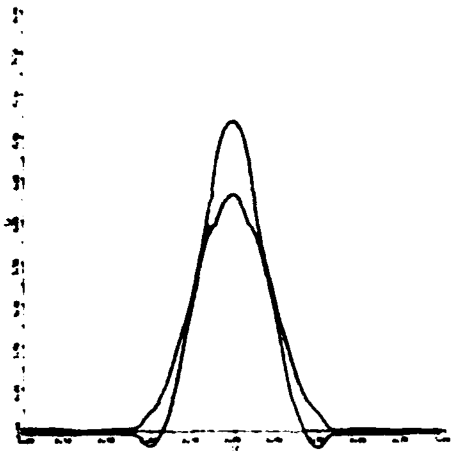


Figure 2

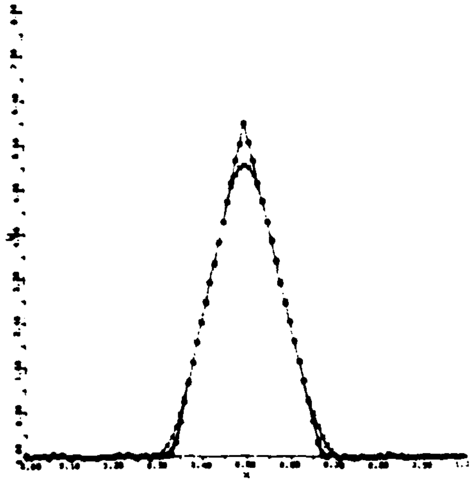


Figure 3

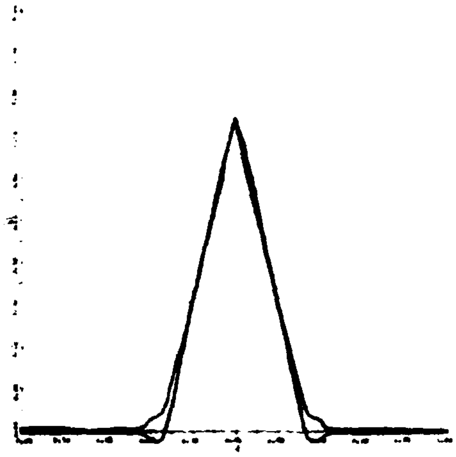


Figure 4

References

Bozzini, M. and Lenarduzzi, L. (1988a), Smoothing with weight variable both in shape and support. To be submitted

Bozzini, M. and Lenarduzzi, L. (1988b), Smoothing with weight variable both in shape and support: numerical aspects. To be submitted

Müller, H. G. and Stadtmüller, U. (1987), Variable bandwidth kernel estimators of regression curves. Ann. Statist. 15, 182-201

Ross, S. M. (1980), 'Introduction to Probability Models', Academic Press, New York, pp 60-61

Silverman, B. W. (1985), Some aspects of the spline smoothing approach to non-parametric regression curve fitting. J. Roy. Statist. Soc. B, 47, 1-52

Wahba, G. (1983), Bayesian confidence intervals for the cross validated smoothing spline. J. Roy. Statist. Soc. B, 45, 135-150

5. Complex Approximation

NUMERICAL METHODS FOR CHEBYSHEV APPROXIMATION OF COMPLEX-VALUED FUNCTIONS

G A WATSON

Department of Mathematics and Computer Science

University of Dundee

Abstract This paper is concerned with the problem of best Chebyshev approximation of a given complex function on a contour of the complex plane. The emphasis is on the provision of methods which can combine global convergence with rapid local convergence, and some different approaches are examined, with particular attention being paid to the special case when the approximating function is linear. Examples are used to illustrate various situations which can arise, and numerical results are presented to demonstrate the perfomance of the methods on a variety of linear problems.

Keywords: Chebyshev approximation, complex functions, algorithms, global convergence.

1. Introduction

Let Γ be a contour of the complex plane C, and let $r:C^n \times \Gamma \rightarrow C$ be an analytic function. Then of interest here is the problem:

$$\text{find } \mathbf{z} \in C^n \text{ to minimise } \|r(\mathbf{z},.)\|, \qquad (1)$$

where $\|r(\mathbf{z},.)\| = \max_{w \in \Gamma} |r(\mathbf{z},w)|$, and $\mathbf{z} = (z_1, z_2, \ldots, z_n)^T \in C^n$. Such problems frequently arise from best approximation problems defined on a simply connected region of the complex plane with boundary Γ, for the use of the maximum modulus theorem means that attention may be restricted to the boundary. For $j = 1,2,...,n$, let $\phi_j(\mathbf{z},w)$ denote the partial derivative of r with respect to z_j. By considering the real problem equivalent to (1) (i.e. with $2n$ real variables formed by taking the real and imaginary parts of z), the following result may be established (see, for example, Ben-Tal et al (1979)), giving necessary conditions for a solution to (1).

<u>Theorem 1</u> Let $\mathbf{z} \in C^n$ solve (1). Then there exist $m \leq 2n+1$ points $w_1, w_2, \ldots, w_m$ in Γ, with

$$|r(\mathbf{z},w_j)| = \|r(\mathbf{z},.)\|, \quad j=1,2,\ldots,m,$$

and λ_j, $j=1,2,\ldots,m$ real nonnegative numbers (not all zero) such that

$$\sum_{j=1}^{m} \lambda_j \overline{r(\mathbf{z},w_j)} \phi_k(\mathbf{z},w_j) = 0, \; k=1,2,\ldots,n.$$

Remark For the special case when r is an affine function of z, it follows that the conditions of the theorem are also sufficient for a solution, and therefore represent characterization conditions (see also Lorentz(1966)). Otherwise the conditions may be said to define a stationary point.

Assume that Γ can be parameterized by the real number $t \in [\alpha,\beta]$ so that $w:[\alpha,\beta] \rightarrow \Gamma$ and $w_j = w(t_j)$, $j=1,2,\ldots,m$. Define $f: R^{2n} \times R \rightarrow R$ by

$$f(\mathbf{x},t) = |r(\mathbf{z},w(t))|^2, \quad t \in [\alpha,\beta], \mathbf{z} \in C^n,$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{z}^R \\ \mathbf{z}^I \end{bmatrix} \in R^{2n}$, and the superscripts R and I denote real and imaginary parts respectively. Without loss of generality, assume that $t_1, t_2, \ldots, t_p$ are internal to $[\alpha,\beta]$. Then if m and p are known, the conditions of the theorem may be written as the nonlinear system of equations

$$\sum_{j=1}^{m} \lambda_j \nabla f(\mathbf{x},t_j) = 0, \tag{2a}$$

$$1 - \sum_{j=1}^{m} \lambda_j = 0, \tag{2b}$$

$$f'(\mathbf{x},t_j) = 0, \quad j=1,2,\ldots,p, \tag{2c}$$

$$f(\mathbf{x},t_j) - h = 0 \;,\; j=1,2,\ldots,m, \tag{2d}$$

where ∇ denotes differentiation with respect to $\mathbf{x} \in R^{2n}$, the dash denotes differentiation with respect to t and the fact is used that

$$\nabla f(\mathbf{x},t) = 2 \begin{bmatrix} Re(\overline{r}\phi) \\ -Im(\overline{r}\phi) \end{bmatrix} \in R^{2n},$$

with $\phi = (\phi_1, \phi_2, \cdots, \phi_n)^T$. This is a system of $2n+m+p+1$ real equations for the $2n+m+p+1$ real unknowns $\mathbf{x}\in R^{2n}, \lambda_j \in R$, $j=1,2,..,m$, $t_j \in R$, $j=1,2,...,p$, and $h\in R$, where $h=\|r(\mathbf{z},.)\|^2$ at a solution to (1).

2. A locally convergent method

The Jacobian matrix of the system (2) may be written in the form

$$\begin{bmatrix} H & 0 & A^T\Lambda & G^T \\ 0 & 0 & 0^T & -\mathbf{e}^T \\ A & 0 & D & 0 \\ G & -\mathbf{e} & \begin{bmatrix} D_1 \\ 0 \end{bmatrix} & 0 \end{bmatrix} \tag{3}$$

where

$$H = \sum_{j=1}^{m} \lambda_j \nabla^2 f(\mathbf{x}, t_j) \in R^{2n\times 2n},$$

$$\Lambda = \text{diag}\ \{\lambda_1, \lambda_2, \ldots, \lambda_p\},$$

$$A_{jk} = \frac{\partial^2 f(\mathbf{x}, t_j)}{\partial t\, \partial x_k},\ j=1,2,...,p,\ k=1,2,...,2n,$$

$$G_{jk} = \frac{\partial f(\mathbf{x}, t_j)}{\partial x_k},\ j=1,2,...,m,\ k=1,2,...,2n,$$

$$D = \text{diag}\ \{f''(\mathbf{x}, t_j),\ j=1,2,..,p\},$$

$$D_1 = \text{diag}\ \{f'(\mathbf{x}, t_j),\ j=1,2,..,p\},$$

and $\mathbf{e}$ denotes the vector in R^m all of whose components are 1. Notice that D is negative semi-definite at a solution to (2) which is also a solution to (1). Also

$$\nabla^2 f(\mathbf{x}, t) = 2\begin{bmatrix} Re(\phi\overline{\phi}^T) & Im(\phi\overline{\phi}^T) \\ -Im(\phi\overline{\phi}^T) & Re(\phi\overline{\phi}^T) \end{bmatrix} + 2\begin{bmatrix} Re(r\overline{T}) & Im(r\overline{T}) \\ Im(r\overline{T}) & -Re(r\overline{T}) \end{bmatrix}, \tag{4}$$

where T is the Hessian matrix of r with respect to $\mathbf{z}$.

The system of equations defining the Newton step is therefore an unsymmetric $2n+m+p+1$ by $2n+m+p+1$ system of equations. Considerable improvement may be obtained if use is made of the fact that D_1 is zero at a solution to (1), so that it may be replaced by zero in (3) without inhibiting a second order convergence rate. The system of equations defining the increment vector may then be written

$$\begin{bmatrix} H & 0 & A^T\Lambda & G^T \\ 0 & 0 & 0 & -\mathbf{e}^T \\ A & 0 & D & 0 \\ G & -\mathbf{e} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta h \\ \delta\mathbf{t} \\ \delta\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -G^T\boldsymbol{\lambda} \\ \boldsymbol{\lambda}^T\mathbf{e}-1 \\ -\mathbf{f}' \\ h\mathbf{e}-\mathbf{f} \end{bmatrix}, \qquad (5)$$

where $\mathbf{f}' \in R^p$ denotes the vector with jth component $f'(\mathbf{x}, t_j)$, $j=1,2,...,p$, and $\mathbf{f} \in R^m$ denotes the vector with jth component $f(\mathbf{x}, t_j)$, $j=1,2,..,m$. This system of equations may readily be broken down into smaller systems. Indeed the following algorithm may be used which first computes a first order estimate of $\boldsymbol{\lambda}$ (an approximation to $\boldsymbol{\lambda}+\delta\boldsymbol{\lambda}$ satisfying (5)) and then provides approximations to $\delta\mathbf{x}$, δh, $\delta\mathbf{t}$ satisfying (5). When used as increment vectors, a second order convergence rate is not inhibited (see, for example Watson(1985)).

Algorithm 1

Step 1 Let $C^T = \begin{bmatrix} G^T \\ -\mathbf{e}^T \end{bmatrix} = [\ Y : Z\ ] \begin{bmatrix} R \\ 0 \end{bmatrix} = Y\,R$,

where $[Y : Z]$ is $(2n+1)\times(2n+1)$ orthogonal and R is $m\times m$ upper triangular, nonsingular if C has full rank (a necessary condition for the system (5) to be nonsingular).

Step 2 Determine $\boldsymbol{\lambda}$ satisfying $R\boldsymbol{\lambda} = -Y^T\mathbf{e}_{2n+1}$ ($\mathbf{e}_{2n+1}$ is thc $(2n+1)^{th}$ co-ordinate vector) and scale $\boldsymbol{\lambda}$ so that $\boldsymbol{\lambda}^T\mathbf{e} = 1$.

Step 3 Solve $R^T\mathbf{d}_1 = h\mathbf{e}-\mathbf{f}$.

Step 4 Form $W = H - A^T\Lambda D^{-1}A$, and solve

$$Z_1^T W Z_1 \mathbf{d}_2 = Z_1^T(\ A^T\Lambda D^{-1}\mathbf{f}' - WY_1\mathbf{d}_1\ ) - \mathbf{z}_2,$$

where Y_1, Z_1 denote the first $2n$ rows of Y, Z respectively, and $\mathbf{z}_2^T$ denotes the last row of Z.

Step 5 Form $\begin{bmatrix} \delta\mathbf{x} \\ \delta h \end{bmatrix} = Y\mathbf{d}_1 + Z\mathbf{d}_2$ and set $\mathbf{x} = \mathbf{x}+\delta\mathbf{x}$, $h = h+\delta h$

Step 6 Form $\delta\mathbf{t} = -D^{-1}(\mathbf{f}'+A\,\delta\mathbf{x})$ and (unless convergence) set $\mathbf{t} = \mathbf{t}+\delta\mathbf{t}$ and return to Step 1.

This algorithm will be dominated by the calculation of the QR factorization of C^T in Step 1. It can converge to a solution of (1) if provided with good initial approximations, and with the correct information about m and p. Of course even if convergence is obtained it has still to be confirmed that $h = \|f\|$, and this is a nontrivial problem in itself. Its solution is connected with a modification of this method in which the vector $\mathbf{t}$ at each iteration is defined to be the appropriate set of m local maxima of the current f in $[\alpha,\beta]$: this calculation can be done as a preliminary step, with Step 6 of course no longer required. Provided that D is nonsingular, and $f'(\mathbf{x},t_j)\neq 0, j=p+1,..,m$ the implicit function theorem can be used to give $\mathbf{t}$ as a differentiable function of $\mathbf{x}$, and the Newton step (in the variables $\mathbf{x},h,\lambda$) derived. The analogue of Algorithm 1 is then obtained by introducing Step 0 as described, dropping Step 6 (testing for convergence in Step 5) and setting $\mathbf{f}'$ to zero in Step 4: a second order convergence rate can again be established.

The following examples all use Algorithm 1 in its original form.

Example 1 (Glashoff and Roleff(1981)) Consider the approximation of w^8 by a complex polynomial of degree 7 on the rectangle $[-2,2] \times [-1,1]$ in the complex plane. Because of symmetry, it is sufficient to consider the approximation of w^8 by a *real* linear combination of $1,w^2,w^4,w^6$ on

$$w(t) = \begin{cases} 2+it, & 0\leq t\leq 1 \\ (3-t)+i, & 1\leq t\leq 3 \end{cases} .$$

The fact that the coefficients are real means that H,A and G may all be reduced in size with consequent reduction in the computation involved in Algorithm 1. Taking the initial approximation $\mathbf{z} = \{(-15.19822,0), (46.28396,0), (-31.88876,0), (7.748537,0)\}$, $\mathbf{t} = (0.325, 1.8, 1.0, 3.0)$, with $m = 4$ and $p = 2$ (interpreting $t_3 = 1.0$ as an end point), and $h = 10428.39$, the performance of the algorithm is summarized in Table 1. The column headed k gives the iteration number and δ gives the maximum modulus component of $\delta\mathbf{x},\delta h$ and $\delta\mathbf{t}$. The CPU time (DEC 10 single precision) was 0.78 seconds. (Notice that the size of the matrix (3) in this case is 11 × 11, whereas the corresponding Jacobian matrix in the method used by Glashoff and Roleff is 15 × 15).

k	δ	t_1	t_2	h
1	0.149281	0.315838	1.794251	10428.54
2	0.479269	0.316292	1.795387	10429.02
3	0.008201	0.316291	1.795388	10429.03
4	0.000025	0.316291	1.795388	10429.03

Table 1

Example 2 Consider the approximation of $1/(w-(2+i))$ by a polynomial of degree 7 on the part of the unit circle defined by $\theta \in [0,3\pi/2]$. In this case m=9 and p=7. Taking the initial approximation z = {(–0.4,0.2), (-0.1202,0.1602) , (–0.01653,0.08797), (0.01073,0.03786), , (0.001226,0.01225), (0.008144,0.002251), (0.004343,–0.00308), (0.001954,–0.000459)}, t = (0,0.3436 ,0.8467, 1.399, 2.0985, 2.815, 3.607, 4.3196, $3\pi/2$) and h=0, the algorithm performed as shown in Table 2. The CPU time was 2.71 seconds.

k	δ	$h \times 10^{-7}$
1	0.04765	4.0788
2	0.04300	4.3852
3	0.01111	4.44382
4	0.000362	4.44747
5	0.000005	4.44755

Table 2

Example 3 This is the approximation of $exp(w)$ on the part of the unit circle defined by $\theta \in [0,\pi]$ by a rational function formed by the quotient of two complex polynomials of degree 2: the rational function is normalized by choosing the constant in the denominator to be 1. In this case m = 6 and p = 4. Taking the initial approximation z = {(1.0066,0.0003384), (0.51337,0.039861), (0.085256, 0.01995), (–0.48537, 0.037976), (0.073634, -0.015799)}, (with the natural ordering of the coefficients), t = { 0, 7α, 22α, 42α, 65α, π }, where α =π/81, and h=0.0, the algorithm performed as summarized in Table 3.

k	δ	$h \times 10^{-7}$
1	0.02148	1.69225
2	0.01872	1.77062
3	0.000452	1.77476
4	0.000002	1.77477

Table 3

The question remains: how can information about m and p, and also good approximations to the unknowns, be provided? It should be emphasised that extremely good initial approximations may be required: for instance in Example 3, small perturbations of the initial values can lead to divergence. In particular, very good approximations to $\mathbf{t}$ are often necessary: there is more about this later on. Perhaps the most obvious way of finding initial data is from the solution (or approximate solution) of a discretization of (1), where Γ is replaced by a finite discrete subset, and this gives rise to a class of two-phase methods. Algorithms of this type have been suggested for linear problems (when r is is an affine function of $\mathbf{z}$) by, for example, Glashoff and Roleff (1981), Grothkopf and Opfer (1982), Opfer (1982) and Mason and Opfer(1986), based on the formulation of the discrete problem as a semi-infinite linear programming problem. Linear programming methods may be used to approximately solve this problem, and advantage can be taken of the special structure to make this efficient (see, for example, Streit (1985, 1986), Streit and Nuttall (1982,1983)). Notice that this procedure requires a double discretization of the original problem (1).

An alternative first phase procedure for the linear problem is suggested by Watson(1988) where the discretization of problem (1) is (directly) solved as a nonlinear programming problem. Advantage is taken of the special form of the problem so that a second order convergent method is obtained, and computational efficiency is achieved by a procedure which restricts attention to small subsets of the original set of discrete points. This method may be adapted to apply to nonlinear problems, although since the convex nature of the problem is lost, the provision of a satisfactory algorithm represents a nontrivial modification and the implementation details have still to be resolved. Some success has been obtained through the use of a Gauss-Newton type of approach (the simple device of ignoring the second term on the right hand side of (4)), and this was used to generate the initial data for Example 3.

Of course both linear and nonlinear discretized problems are examples of nonlinear minimax optimization problems, and standard methods for such problems are available. The main computational effort is expended in identifying the correct index set on which a solution is defined (analogous to the points $w_1, w_2, \ldots, w_m$ of Theorem 1), and most methods are active set methods which iterate towards a correct index set through equality constrained quadratic programming subproblems defined on estimates of this set. Recent developments have been concerned with making better use of the special structure and solution characterization of minimax problems, and for example a method having this goal is given by Yuying Li(1988). A particular feature of the problems considered here is that fine discretizations (and therefore a large number of points) may be necessary, and the relative effectiveness of different approaches has yet to be studied. In particular, the treatment of nonlinear problems of the type (1) and their discretizations has received little attention (with the exception of some rational approximation problems) and so is not something that can be properly dealt with here. For the rest of this paper, therefore, although much of the theory carries over (or may be modified to apply) to the nonlinear case, attention will be confined to the treatment of the linear problem for which

$$r(\mathbf{z}, w) = \sum_{j=1}^{n} z_j \phi_j(w) - F(w), \quad w \in \Gamma, \tag{6}$$

where $F:\Gamma \to C, \phi_j:\Gamma \to C$, $j=1,2,\ldots,n$ are analytic functions. Perhaps the most important consequence of this is that $\nabla^2 f(\mathbf{x}, t)$ is given by the first term on the right hand side of (4) and so is positive semi-definite; therefore so also are H and W provided that the components of λ are non-negative. H is of course independent of $\mathbf{x}$.

3. Globalization

The problem (1) may be stated in the form

$$\begin{gathered} \text{minimize } h \\ \text{subject to } f(\mathbf{x}, t) \le h, \quad t \in [\alpha, \beta], \end{gathered} \tag{7}$$

and a corresponding discrete problem may be written

$$\begin{gathered} \text{minimize } h \\ \text{subject to } f(\mathbf{x}, t_i) \le h, \quad i=1,2,\ldots,s, \end{gathered} \tag{8}$$

where $t_i \in [\alpha, \beta]$, $i=1,2,\ldots,s$. Clearly if the discrete set contains the points t_i defined by Theorem 1 then the minimum values of both problems will be the same. One approach to solving (7) is therefore to iterate on a sequence of discrete subsets which in the limit contains the appropriate optimal set.

Given $\mathbf{x} \in R^{2n}$ let $t_1, t_2, \ldots, t_q$ be the set of local maxima (assumed finite) of $f(\mathbf{x}, t)$ in $[\alpha, \beta]$. Then

$$f'(\mathbf{x}, t_j) = 0, \quad t_j \in (\alpha, \beta),$$

and provided that $f''(\mathbf{x}, t_j) \ne 0$, $t_j \in (\alpha, \beta)$, $f'(\mathbf{x}, t_j) \ne 0$, $t_j \in \{\alpha, \beta\}$, the implicit function theorem may be applied to give t_j, $j=1,2,\ldots,q$ as differentiable functions of $\mathbf{x}$. Based on these q local maxima, and given $\lambda \in R^q$ non-negative , define $\mathbf{f} \in R^q$, $G \in R^{q \times 2n}$, $H \in R^{2n \times 2n}$, $W \in R^{2n \times 2n}$,$C \in R^{q \times (2n+1)}$ as in Algorithm 1. Let $h = \max_{1 \le j \le q} f(\mathbf{x}, t_j)$ and consider the quadratic programming problem

$$\text{minimize } p+\tfrac{1}{2}\mathbf{d}^T W\mathbf{d}$$

$$\text{subject to } G\mathbf{d}-p\mathbf{e} \le h\mathbf{e}-\mathbf{f}. \qquad (9)$$

<u>Theorem 2</u> For r given by (6), let p,$\mathbf{d}$ solve (9) defined at $\mathbf{x}$. Then

(a) if $p = 0$, $\mathbf{x}$ solves (7),

(b) if $p < 0$, $\mathbf{d}$ is a descent direction for $\|f(\mathbf{x}\,,\,.\,)\|$ at $\mathbf{x}$.

<u>Proof</u> By assumption

$$W = H-A^T\Lambda D^{-1}A$$

is positive semi-definite if it is defined. The Kuhn-Tucker conditions imply the existence of a non-negative vector $\mu\in R^q$ such that

$$\mathbf{e}^T\mu = 1,$$

$$W\mathbf{d} + G^T\mu = 0.$$

Thus

$$\mathbf{d}^T W\mathbf{d} + \mathbf{d}^T G^T\mu = 0,$$

so that $\mathbf{d}^T G^T \mu \le 0$. Also

$$\mu^T(h\mathbf{e}-\mathbf{f}-G\mathbf{d}+p\mathbf{e}) = 0,$$

implies $h+p-\mu^T\mathbf{f} = \mu^T G\mathbf{d} \le 0$, so that $p \le \mu^T\mathbf{f}-h \le 0$.

If p=0, $\mathbf{d}^T W\mathbf{d} = 0$, so that if $\mathbf{d}\neq 0$, there is another solution $p = 0$, $\mathbf{d} = 0$, and the Kuhn-Tucker conditions are then equivalent to (2) being satisfied.

Now assume $p < 0$. We may write

$$f_j(\mathbf{x}) = f(\mathbf{x},t_j(\mathbf{x})) \quad j=1,2,\ldots,q,$$

showing the dependence of t_j on x. Further for γ>0 small enough

$$\|f(\mathbf{x}+\gamma\mathbf{d},\,.)\| = \max_{1\le j\le q} f_j(\mathbf{x}+\gamma\mathbf{d}).$$

Now define J by

$$h = f_j\,,\ j\in J.$$

Then $\mathbf{d}^T\nabla f_j \le p < 0$, $j \in J$, so that

$$f_j(\mathbf{x}+\gamma\mathbf{d}) = f_j(\mathbf{x})+\gamma\mathbf{d}^T\nabla f_j+0(\gamma^2)$$

$$<f_j(\mathbf{x})$$

for $\gamma > 0$ small enough. It follows that

$$\|f(\mathbf{x}+\gamma\mathbf{d}, .)\| < \max_{1\le j\le q} f_j(\mathbf{x}) = \|f(\mathbf{x}, .)\|,$$

and the result is proved.

Remark This result goes through for nonlinear problems provided that W is a positive semi-definite matrix (or is replaced by a positive semi-definite matrix).

The new point $\mathbf{x}+\gamma\mathbf{d}$ may be obtained by a line search, and it is usual to choose γ to satisfy the inequality

$$\|f(\mathbf{x}+\gamma\mathbf{d}, .)\| \le \|f(\mathbf{x}, .)\|+\gamma p\tau \ , \qquad (10)$$

where τ is a small positive number, for example 0.0001, and γ is large in $[0,1]$, with the value 1 chosen if possible. Then the following result may be established.

Theorem 3 Let $\{\mathbf{d}^k\},\{p^k\}$ be sequences defined by solutions of (9) at $\mathbf{x}^k$ using (10), where the superscripts k on other quantities imply evaluation at $\mathbf{x}^k$. Then if $\{W^k\},\{\mathbf{d}^k\}$ are bounded, the limit points of $\{\mathbf{x}^k\}$ solve (7).

Proof The result that $\{p^k\}\rightarrow 0$ may be established by the application of standard techniques available in optimization theory.

Going to a subsequence if necessary, let $\{\mathbf{x}^k\}\rightarrow \mathbf{x}^*$. Now, using boundedness, and going to further subsequences if necessary, $\mu^k\rightarrow \mu^*, W^k\rightarrow W^*$, $\mathbf{d}^k\rightarrow \mathbf{d}^*$ as $k\rightarrow\infty$. Thus $\mathbf{d}^*$, $p^*(=0)$ solve (9) at $\mathbf{x} = \mathbf{x}^*, W=W^*$ with objective function value zero. Thus a (possibly different) solution is given by $\mathbf{d} = 0$, $p = 0$ and the conclusion that $\mathbf{x}$ is a solution follows as before.

The following algorithm may be interpreted as being of multiple exchange type.

Algorithm 2

Step 0 Determine the local maxima of f in $[\alpha,\beta]$, say $t_1,t_2,\ldots,t_q$. Set $h = \|f(\mathbf{x}, .)\|$.

Step 1 Solve (9), and unless there is convergence, choose γ so that (10) is satisfied, set $\mathbf{x} = \mathbf{x}+\gamma\mathbf{d}$ and go to Step 0.

If the problem (9) is solved with the constraints fixed as *equalities* , then provided that $\mu \geq 0$ the situation is as before. However the solution may now be obtained in an efficient manner by steps similar to those used in Algorithm 1. An alternative exchange algorithm may therefore be defined as follows.

Algorithm 2a

Step 0 As in Algorithm 2.

Steps 1-5 As in Algorithm 1 (with $\mathbf{f}, G$, etc. defined on the current set of local maxima $t_1,t_2, \cdots t_q$), except that we can set $\mathbf{f}' = 0$ in Step 4, and a full step is not necessarily taken in Step 5 for the new $\mathbf{x}$.

Although the steps of Algorithm 2a are computationally more efficient (and the connection with Algorithm 1 immediately gives a desirable local property), there is no guarantee that all the components of the multiplier vector will be non-negative at solutions to (the modified) problem (9), in particular far from a solution to (1). Therefore an active set strategy (for handling negative μ_j) is required for a practically useful (globally convergent) algorithm. On the other hand, the solution of the inequality constrained problem (9) automatically picks out the correct active set in the event that not all constraints are active. Eventually, the two algorithms should be effectively the same, so that a second order convergent rate is normal with Algorithm 2. An important part of both these algorithms is the calculation of all the local maxima: this is needed not just for Step 0, but also for the implementation of the step length test. The process is not a finite one, and typically is achieved in two stages: firstly a grid search to approximately identify the locations of the maxima; secondly a local procedure (for example Newton's method) to give more accurate values. Incidentally, this is an area in which the availability of a parallel computing facility could lead to great benefits.

An exchange method of either kind can be applied to solve *discretizations* of continuous problems, with fairly obvious modifications to the algorithms (reflecting the absence of (2c) from the conditions to be satisfied): local maxima on the finite set *only* are used, $W = H$ in Step 4 of Algorithm 1 and also in (9) and the first term on the right hand side of the system of equations in Step 4 is absent. However there are potential difficulties, if the number of extrema of discrete and continuous problems do not match. (In this context *extrema* refers to points where the norm is attained.) For example for the approximation of $F(w) = 1/(w-(2+i))$ by a polynomial of degree 3 in the first quadrant of the unit circle, there are 5 extrema of the continuous solution, with parameter values 0, 0.259415, 0.733998, 1.269364, $\pi/2$. However the solution using the algorithm of Watson on m=101 equispaced points gives rise to 6 extrema, including the neighbouring pair 0.2513 and 0.2670, and taking m = 1001 equispaced points gives 7 extrema, including the triple 0.73042, 0.73356 and 0.73827. For many problems this discrepancy does not occur, and the extent to which it can be attributed to the use of finite precision arithmetic is not clear. Nevertheless, there are two possible dangers associated with this phenomenon: firstly ill-conditioning because of the proximity of active points, and secondly failure of the algorithm through failure to properly locate all of these relevant local maxima as the computation proceeds. If the situation is an inherent property of the problem, it parallels a phenomenon which occurs in real linear Chebyshev approximation: if a continuous real problem (with n real coefficients) is discretized, then in the full rank case there is always a solution with (n+1) extrema, and this is the solution which is obtained if a standard technique is applied. However for non-Chebyshev set problems, the solution to the original continuous problem (assumed unique) may have fewer than (n+1) extrema: such problems have been referred to as *singular* by Osborne and Watson(1969). A difference in the complex case is that it does not seem possible to predict in advance the number of extrema of discretized problems, and indeed this appears to depend on the particular discretization. Nevertheless the same interpretation of *singularity* may be made.

The above phenomenon has no bearing on the direct solution of the continuous complex problem by exchange methods of the type described above, and Algorithm 2 was applied to a number of different problems. Far from the solution, convergence could be slow and two modifications were found to be beneficial : firstly, W was modified to $W+\mu I$, for some $\mu \geq 0$, with μ being allowed to decrease as the algorithm approaches the solution; secondly in the early stages only local maxima on a discrete set were determined (and so W was chosen equal to H), with accurate local maxima obtained only when p became sufficiently small.

Example 4 This is the approximation of w^3 by a polynomial of degree 2 on the first octant of the unit circle. Initially local maxima were sought on a grid of 101 equispaced points. The initial approximation was given by $z_j = (0,0)$, j=2,3, $z_1 = (1,0)$ and the performance of the method is summarized in Table 4. The CPU time was 6.63 seconds.

k	q	μ	p	γ	$\|r\|$
1	1	1.0	-5.85	1	1.274
2	2	0.5	-1.60	1	0.647
16	4	0.0001	-1.6E-5	1	0.014709
17	4	0	-8. E-8	1	0.014708
18	4	0	-2. E-8	1	0.0147077
19	4	0	-1. E-8	1	0.0147077

Table 4

An alternative to the above method is to revert to a 2-phase approach, with algorithm 2 (or indeed Algorithm 2a) as a second phase. This makes the 2-phase method more robust (at the expense of extra computation) and permits convergence to a solution of (7) even if incorrect information is supplied at the end of the first phase, or poor approximations only are available. It also means, however, that while global convergence properties are important, the main emphasis can be on local behaviour; if Algorithm 2 is applied with all constraints initially set to equalities, then frequently no change is required to the active set, so little loss in efficiency results in using Algorithm 2 as opposed to Algorithm 2a. In addition W may be used without modification, and $\gamma = 1$ is the expected value satisfying (10). Therefore the next algorithm tested was based on combining the method of Watson with Algorithm 2. In fact a version of the first method was used where solutions were obtained on successive discrete sets defined as follows. Initially $2n+1$ equispaced points were used. Subsequently each set consisted of the current set of extreme points supplemented by all the local maxima of f on the original discrete point set. When a sufficiently small value of p was reached, or a sufficiently small increment vector was obtained, Algorithm 2 was entered.

Algorithm 3

First Phase The algorithm of Watson, modified as described in the previous paragraph.

Second Phase Algorithm 2, with all constraints of (9) initially set to active.

Both phases of this algorithm involve the solution of a sequence of quadratic programming problems, so that in addition to giving a desirable combination of global and rapid local convergence, they also make use of the same software. At each iteration of both phases, a particular subset of discrete points from $[\alpha,\beta]$ is identified and the constraints of the quadratic programming problem are defined by this subset: in the first phase, this remains constant for several iterations, and not all constraints are normally active; in the second phase, the subset changes at each iteration, and all constraints are usually active.

Consider Example 4 solved by Algorithm 3 (from the same initial approximation). Table 5 gives results of the application of the above algorithm applied to the discretized problem. The number k gives the outer iteration count (the number of times a discrete solution is obtained on a *fixed* discrete subset of the original discrete set), q is the number of quadratic programming subproblems solved at each outer iteration, $\|\mathbf{r}_k\|$ is the value reached of the norm on the current discrete set, and $\|\mathbf{r}\|$ is the norm evaluated on all 101 points. The number l is the number of local maxima added in at each step, and p denotes the value of p reached on termination. The data input to Algorithm 2 was:

$$\mathbf{z} = \{(0.3679780,0.8883755), (-1.988566,-1.988564), (2.630987,1.089790)\}.$$

After 2 iterations, the maximum modulus component of the increment vector was less than 0.00001 and the solution obtained was

$$\mathbf{z}^* = \{ (0.368115,0.888709), (-1.989059,-1.989059), (2.631323,1.089929)\},$$

$$\mathbf{t}^* = (0,0.198232,0.587165,\tfrac{\pi}{4}),$$

$$\lambda^* = (0.167767,0.332231,0.332236,0.167765),$$

with $\|\mathbf{r}^*\| = 0.147077$. The total CPU time was 7.98 seconds.

k	q	$\|\mathbf{r}_k\|$	$\|\mathbf{r}\|$	l	p
1	14	0.011666	0.022209	3	-2.0 -7
2	4	0.014607	0.014865	2	-7.1 -10
3	3	0.014706			-4.6 -9

Table 5

For approximation problems defined on the whole of the unit circle, it is not uncommon for there to be a *constant* error curve, in other words for the minimum norm value to be attained at all points on the unit circle. Studies of this phenomenon for polynomial (and rational) approximation have been made by, for example, Trefethen (1981a,1981b). In this case it is clear that Algorithm 2 will eventually break down; on the other hand , for methods of two-phase type, it is easy to identify this situation at the conclusion of the first phase.

Example 5 Let $f(w) = 1/(w-2)$ be approximated by a polynomial of degree 6 on the whole of the unit circle. The performance of the algorithm of Watson(1988) applied on 100 equispaced points from $\mathbf{z} = \mathbf{e}_1$ is summarised in Table 6. At the end of the second iteration $\|\mathbf{r}\| = 0.005208$ attained at all 100 points, and the conclusion that the error curve is a constant on the unit circle follows.

k	q	$\|\mathbf{r}_k\|$	l
1	10	0.002905	7
2	5	0.005208	

Table 6

The possibility of this kind of behaviour points the way to another source of difficulty in accurately computing solutions to continuous complex Chebyshev approximation problems: although the error curve is not constant, it may be *close* to being constant. This means that identification of the positions of the points where the norm is attained may be quite difficult, and the positions of the local maxima as the computation proceeds will be very sensitive to changes in the coefficient values.

Example 6 Consider the approximation of $f(w) = 1/(1+(w+1)^2)^{\frac{1}{2}}$ by $\sum_{j=1}^{n} z_j(1+w)^{1-j}$ on $\{w : w = iy, -20 \le y \le 20\}$. This is an another example of a complex problem which has a solution for which $\mathbf{z}$ is real, and advantage can be taken of this to reduce the size of the computations: in particular symmetry can be used to restrict consideration to non-negative values of y. In Table 7a is shown the result of applying the algorithm as before on 51 equispaced points on $0 \le y \le 20$, taking $n = 4$ and starting from $\mathbf{z} = \mathbf{e}_1$. The minimum value of $\|\mathbf{r}\|$ is 0.016046 attained at the 5 points defined by $\mathbf{t} = (0.4, 0.8, 1.6, 2.0, 20.0)^T$. The points corresponding to local **minima** of $|r_i|$ are 0.0, 1.2 and 3.6 with values 0.015983, 0.016036 and 0.016039 respectively. Table 7b shows the performance of Algorithm 2 applied from the discrete solution; the continuous problem has just 3 extrema and the sensitivity of the local maxima to changes in the coefficients is clearly demonstrated. The total CPU time was 7.81 seconds.

k	q	$\|\mathbf{r}_k\|$	$\|\mathbf{r}\|$	l
1	16	0.000891	0.086371	1
2	8	0.014233	0.024608	1
3	4	0.015954	0.017687	2
4	3	0.016041	0.016078	2
5	2	0.016045	0.016052	1
6	2	0.016046		

Table 7a

k	t_1	t_2	t_3	δ	$\|r\|$
1	0.571362	1.777237	20.0	0.000102	0.016059
2	0.334464	1.650488	20.0	0.000034	0.016051
3	0.439621	1.610323	20.0	0.000023	0.016049
4	0.465623	1.639444	20.0	0.000002	0.016049

Table 7b

Finally the performance of Algorithm 3 applied to some other examples is summarized in Table 8. The different functions used were

$$f_1(w) = (sinw/w)^{\frac{1}{2}},$$

$$f_2(w) = (w-(2+i))^{-1},$$

$$f_3(w) = (1+(w+1)^2)^{-\frac{1}{2}},$$

and other information defining the problems being solved is given in the table. All initial approximations were $z = e_1$. The number δ gives the maximum modulus component of the increment vector on termination, p denotes the final value of p, $\|r\|$ is the final value of the norm, and CPU denotes the CPU time in seconds.

$f(w)$	$\phi_j(w)$	w	α	β	n	$\delta\times10^{-5}$	$p\times10^{-9}$	$\|r\|$	CPU
$f_1(w)$	w^{j-1}	e^{it}	0	$\frac{\pi}{2}$	4	.5	-2.0	0.000281	4.02
$f_2(w)$	w^{j-1}	e^{it}	0	$\frac{\pi}{2}$	4	5.0	-4.0	0.005028	15.86
			0	$\frac{3\pi}{2}$	6	.3	-10.0	0.003863	27.71
			0	$\frac{3\pi}{2}$	8	2.0	-4.0	0.000670	85.89
$f_3(w)$	$(1+w)^{1-j}$	it	0	20	4	.2	-3.0	0.000281	7.81
			0	20	6	.6	-0.2	0.000140	13.70
	$(w+j)^{-1}$	it	0	20	6	1.0	-7.0	0.003809	19.63

Table 8

4. Concluding remarks

Some numerical methods have been presented for solving complex Chebyshev approximation problems defined on a contour of the complex plane. It has been assumed that a one-dimensional parameterization of the contour is available, and this has been exploited to increase efficiency of the methods, although it is not a limiting factor. Numerical results have been presented to demonstrate the effectiveness of different approaches for a variety of linear problems, and examples have been used to illustrate various points. The emphasis has been on the provision of methods which can combine global convergence with satisfactory rapid local convergence and it seems that this is best achieved within the context of a two-phase approach: this is characterized by the use of the solution to a discretization of the original problem (and other information produced) as input to a method with (primarily) good local properties for satisfying the characterization conditions. Conventional methods of this type do not monitor progress in the second phase (and in particular are not descent methods), and rely on the input data being sufficiently good that convergence to a solution of the original problem is achieved without difficulty.

The intention here has been to provide a measure of satisfactory progress in the second phase, so that (at the expense of additional computation) convergence to the solution can be more reasonably guaranteed. While the numerical experience is still somewhat limited, it is hoped that the present work will assist in the quest, and provide a satisfactory framework, for good numerical methods for solving linear complex Chebyshev approximation problems.

Finally, it is clear that nonlinear problems may be tackled by the methods presented here. The second phase (in theory) presents no difficulty, and so the main requirement is for a satisfactory method for solving the discretized problem, and this involves, in particular, deciding how to deal with the Hessian matrix of f. One possibility is to ignore that part of the Hessian matrix which requires second derivatives of r, (the second term on the right hand side of (4)) leading to a method of Gauss-Newton type. While this is simple, it is unlikely to be generally effective, and quasi-Newton or finite difference approximations to the Hessian matrix are more attractive propositions. It may be that there is little to be gained by considering alternatives to standard methods for nonlinear minimax problems, particularly if they are tailored to exploit special structure and properties as is done by Yuying Li(1988). This remains to be investigated and will be the subject of future research.

References

Ben-Tal,A., Teboulle,M. and Zowe, J. (1979), Second order necessary optimality conditions for semi-infinite programming problems, in Semi-Infinite Programming, Hettich, R. (ed.), Springer Verlag, Berlin.

Glashoff,K. and Roleff,K. (1981), A new method for Chebyshev approximation of complex-valued functions, Math. Comp. 36, pp.233-239.

Grothkopf, U. and Opfer, G. (1982), Complex Chebyshev polynomials on circular sectors with degree six or less, Math. Comp. 39, pp. 599-615.

Lorentz, G.G. (1966), Approximation of Functions, Holt, Rinehart and Winston, New York.

Mason, J.C. and Opfer, G. (1986), An algorithm for complex polynomial approximation with nonlinear constraints, in Approximation Theory V, Chui, C.K., Schumaker, L.L. and Ward, J. (eds.), Academic Press, New York.

Opfer, G. (1982), Solving complex approximation problems by semi-infinite-finite optimization techniques: a study on convergence, Num. Math. 39, pp. 411-420.

Osborne, M.R. and Watson, G.A. (1969), A note on singular minimax approximation problems, Jour. Math. Anal. and Appl. 25, pp. 692-700.

Streit, R.L. (1985), An algorithm for the solution of systems of complex linear equations in the l_∞ norm with constraints on the unknowns, ACM Trans. Math. Software 11, pp. 242-249.

Streit, R.L. (1986), Solution of systems of complex linear equations in the L_∞ norm with constraints on the unknowns, SIAM J. Sci. Stat. Comp. 7, pp. 132-149.

Streit, R.L. and Nuttall, A.H. (1982), Linear Chebyshev complex function approximation and an application to beamforming, J. Ass. Soc. Amer. 72, pp. 181-190.

Streit, R.L. and Nuttall, A.H. (1983), A note on the semi-infinite programming approach to complex approximation, Math. Comp. 40, pp. 599-605.

Trefethen, L.N. (1981a), Rational Chebyshev approximation on the unit disc, Num. Math. 37, pp. 297-320.

Trefethen, L.N. (1981b), Near-circularity of the error curve in complex Chebyshev approximation, Jour. Approx. Th. 31, pp. 344-367.

Watson, G.A. (1985), Lagrangian methods for semi-infinite programming problems, in Infinite Programming, Anderson, E.J. and Philpott, A.B. (eds.), Springer Verlag, Berlin.

Watson, G.A. (1988), A method for the Chebyshev solution of an overdetermined system of complex linear equations, IMA Jour. Num. Anal. (to appear).

Yuying Li (1988), An Efficient Algorithm for Nonlinear Minimax Problems, Ph D Thesis, University of Waterloo.

A FAST ALGORITHM FOR LINEAR COMPLEX CHEBYSHEV APPROXIMATION

PING TAK PETER TANG
Argonne National Laboratory

Abstract A natural generalization of the Remez algorithm from real approximation to complex is presented. The algorithm is the first quadratically convergent of its kind.

Keywords Chebyshev approximation, Remez algorithm, quadratic convergence.

1 Introduction

Given a complex function F analytic on a specified domain in the complex plane, how can one construct the polynomial, of a prescribed degree, that best approximates F in the Chebyshev sense? Applications for such an algorithm can be found in [2], [3], [5], [6], and [8]. In the past, complex Chebyshev polynomial approximation has been far less well understood than its real analogue. In particular, the quadratically convergent Remez algorithm ([7] and [13]) for real approximation has not been satisfactorily generalized to a quadratically convergent algorithm for complex approximation. In this paper, we describe a natural generalization of the real Remez algorithm to complex that converges quadratically when conditions similar to those for the real case are satisfied.

The organization of this paper is as follows. In Section 2 we formulate our approximation problem in a form that facilitates the generalization. In Section 3, we take a short digression on the real Remez algorithm before the complex Remez algorithm is presented. The convergence properties of the algorithm, both global and local, are discussed in Section 4. Although the assumptions in Section 4 for quadratic convergence are not met very often in practice, we are able to relax them by an extension of the algorithm in Section 5. Finally, some concluding remarks are made in Section 6.

2 Formulation

Consider a complex-valued function F analytic in a domain enclosed by a smooth closed curve in the complex plane. Let $\gamma : [0,1] \rightarrow \mathbf{C}$, $\gamma(0) = \gamma(1)$, be a smooth parametrization of that curve, which is the boundary of the domain. Define $f, \varphi_1, \varphi_2, \ldots, \varphi_{2n}$, and

p as follows:

$$\begin{aligned} F(t) &:= f(\gamma(t)), \\ \varphi_l(t) &:= \mathrm{Re}\Big(\gamma^{l-1}(t)\Big), \quad l = 1,2,\ldots,n \\ \varphi_{n+l}(t) &:= \iota\mathrm{Im}\Big(\gamma^{l-1}(t)\Big), \quad l = 1,2,\ldots,n \\ p(\lambda,t) &:= \sum_{l=1}^{2n} \lambda_l \varphi_l(t) \quad \text{for any } \lambda = [\lambda_1,\lambda_2,\ldots,\lambda_{2n}]^T \in \mathbf{R}^{2n}. \end{aligned}$$

Let $\|\cdot\|$ be the Chebyshev norm over [0,1]. Then the complex Chebyshev approximation problem is to find $h^* \in \mathbf{R}$ and $\lambda^* \in \mathbf{R}^{2n}$ such that

$$h^* = \Big\|f - p(\lambda^*,\cdot)\Big\| \le \Big\|f - p(\lambda,\cdot)\Big\| \quad \text{for all } \lambda \in \mathbf{R}^{2n}.$$

3 A Complex Remez Algorithm

In this section, we first revisit the familiar real Remez algorithm and view it as a simplex algorithm that solves the dual of the approximation problem. Although that dual is a semi-infinite linear program, the method of solution needs no discretization or two different phases, as done in [4] and [10]. Once the real Remez is realized as such, simply doubling its dimension yields an algorithm for the complex problem.

3.1 The real Remez algorithm

For the duration of this subsection, assume that f is real and $\gamma = 1$. Thus $F = f$ and $\gamma_l(t) = t^{l-1}, l = 1,2,\ldots,n$. The approximation problem can be viewed as the following minimization problem: Determine $h^* \in \mathbf{R}$ and $\lambda^* \in \mathbf{R}^n$ so as to minimize h, subject to

$$h \ge e^{-\iota\vartheta}\Big(f(t) - p(\lambda,t)\Big), \quad \text{for all } (t,\vartheta) \in [0,1] \times \{0,\pi\}.$$

The dual [9] of the minimization problem is as follows: Determine an infinite vector $r : \mathcal{I} \to [0,1]$, where $\mathcal{I} := [0,1] \times \{0,\pi\}$, so as to maximize

$$h = \sum_{(t,\vartheta)\in\mathcal{I}} r(t,\vartheta) e^{-\iota\vartheta} f(t)$$

subject to

$$\sum_{(t,\vartheta)\in\mathcal{I}} r(t,\vartheta) = 1, \quad \text{and}$$

$$\sum_{(t,\vartheta)\in\mathcal{I}} r(t,\vartheta) e^{-\iota\vartheta} \varphi_l(t) = 0, \quad l = 1,2,\ldots,n.$$

The infinite sum can be justified since the optimum is achievable by a vector r^* with at most n+1 nonzero entries.

Given $\boldsymbol{t} = [t_1, t_2, \ldots, t_{n+1}]^T \in [0,1]^{n+1}$ and $\boldsymbol{\vartheta} = [\vartheta_1, \vartheta_2, \ldots, \vartheta_{n+1}]^T \in \{0, \pi\}^{n+1}$, define two $n+1$-vectors and an $(n+1) \times (n+1)$ matrix as follows:

$$\begin{aligned} \boldsymbol{b} &:= [1, 0, 0, \ldots, 0]^T \\ \boldsymbol{a}(t, \vartheta) &:= [1, e^{-\imath\vartheta}\varphi_1(t), e^{-\imath\vartheta}\varphi_2(t), \ldots, e^{-\imath\vartheta}\varphi_n(t)]^T \\ A(\boldsymbol{t}, \boldsymbol{\vartheta}) &:= [\boldsymbol{a}(t_1, \vartheta_1), \boldsymbol{a}(t_2, \vartheta_2), \ldots, \boldsymbol{a}(t_{n+1}, \vartheta_{n+1})]. \end{aligned}$$

The one-point exchange Remez algorithm for real approximation can be stated as follows:

Algorithm 1

Step 0 Find an initial $\boldsymbol{t} \in [0,1]^{n+1}$ and $\boldsymbol{\vartheta} \in \{0, \pi\}^{n+1}$ such that $A(\boldsymbol{t}, \boldsymbol{\vartheta})$ is nonsingular and $A^{-1}(\boldsymbol{t}, \boldsymbol{\vartheta}) \cdot \boldsymbol{b} > \mathbf{o}$.

Step 1 Define $\boldsymbol{r}(\boldsymbol{t}, \boldsymbol{\vartheta}) := A^{-1}(\boldsymbol{t}, \boldsymbol{\vartheta}) \cdot \boldsymbol{b}$. From $(\boldsymbol{t}, \boldsymbol{\vartheta})$ determine the unique $\boldsymbol{\sigma} := \begin{bmatrix} h \\ \boldsymbol{\lambda} \end{bmatrix}$ such that

$$e^{-\imath\vartheta}\Big(f(t_j) - p(\boldsymbol{\lambda}, t_j)\Big) = h, \qquad j = 1, 2, \ldots, n+1.$$

Step 2 Determine t' such that $|f(t') - p(\boldsymbol{\lambda}, t')| = \|f - p(\boldsymbol{\lambda}, \cdot)\|$. Define

$$\vartheta' := \begin{cases} 0 & \text{if } f(t') - p(\boldsymbol{\lambda}, t') \geq 0; \\ \pi & \text{otherwise.} \end{cases}$$

Since $h \leq \|f - p(\boldsymbol{\lambda}^*, \cdot)\| \leq \|f(t') - p(\boldsymbol{\lambda}, t')\|$, whenever $\|f(t') - p(\boldsymbol{\lambda}, t')\| - h$ is small, h and $\boldsymbol{\lambda}$ are good approximations to the optimal solution and we terminate the algorithm. Otherwise, we move on.

Step 3 Exchange (t', ϑ') with one of the (t_j, ϑ_j)'s, $j = 1, 2, \ldots, n+1$. This exchange can be determined by solving the following small linear programming problem (via the simplex algorithm): Determine

$$r(t_1, \vartheta_1), r(t_2, \vartheta_2), \ldots, r(t_{n+1}, \vartheta_{n+1}), r(t', \vartheta') \in [0,1]$$

so as to maximize the inner product

$$h = \left(\sum_{j=1}^{n+1} r(t_j, \vartheta_j) e^{-\imath\vartheta_j} f(t_j) \right) + r(t', \vartheta') e^{-\imath\vartheta'} f(t')$$

subject to

$$[A(\boldsymbol{t}, \boldsymbol{\vartheta}) \quad \boldsymbol{a}(t', \vartheta')] \cdot \boldsymbol{r} = \boldsymbol{b}.$$

The optimal basis of this problem is exactly $\{(t_1, \vartheta_1), (t_2, \vartheta_2), \ldots, (t_{n+1}, \vartheta_{n+1})\}$ with one of the elements replaced by (t', ϑ'). Rename the new basis as $(\boldsymbol{t}, \boldsymbol{\vartheta})$. Go back to Step 1.

3.2 The complex Remez algorithm

We are now ready to go back to the complex approximation problem, which can be stated as the following minimization problem: Determine $h^* \in \mathbf{R}$ and $\boldsymbol{\lambda}^* \in \mathbf{R}^{2n}$ so as to minimize h, subject to

$$h \geq \mathrm{Re}\Big(e^{-\iota\vartheta}[f(t) - p(\boldsymbol{\lambda}, t)]\Big), \quad \text{for all } (t, \vartheta) \in [0,1] \times [0, 2\pi].$$

The dual of the minimization problem is as follows: Determine an infinite vector $\boldsymbol{r}$: $\mathcal{I} \rightarrow [0,1]$, where $\mathcal{I} := [0,1] \times [0, 2\pi]$, so as to maximize

$$h = \sum_{(t,\vartheta)\in\mathcal{I}} r(t,\vartheta)\mathrm{Re}\Big(e^{-\iota\vartheta} f(t)\Big)$$

subject to

$$\sum_{(t,\vartheta)\in\mathcal{I}} r(t,\vartheta) = 1, \quad \text{and}$$

$$\sum_{(t,\vartheta)\in\mathcal{I}} r(t,\vartheta)\mathrm{Re}\Big(e^{-\iota\vartheta}\varphi_l(t)\Big) = 0, \quad l = 1, 2, \ldots, 2n.$$

The infinite sum can be justified since the optimum is achievable by a vector $\boldsymbol{r}^*$ with at most $2n + 1$ nonzero entries.

Given $\boldsymbol{t} = [t_1, t_2, \ldots, t_{2n+1}]^T \in [0,1]^{n+1}$ and $\boldsymbol{\vartheta} = [\vartheta_1, \vartheta_2, \ldots, \vartheta_{2n+1}]^T \in [0, 2\pi]^{2n+1}$, define two $2n + 1$-vectors and an $(2n+1) \times (2n+1)$ matrix as follows:

$$\begin{aligned}
\boldsymbol{b} &:= [1, 0, 0, \ldots, 0]^T \\
\boldsymbol{a}(t,\vartheta) &:= [1, \mathrm{Re}(e^{-\iota\vartheta}\varphi_1(t)), \mathrm{Re}(e^{-\iota\vartheta}\varphi_2(t)), \ldots, \mathrm{Re}(e^{-\iota\vartheta}\varphi_{2n}(t))]^T \\
A(\boldsymbol{t},\boldsymbol{\vartheta}) &:= [\boldsymbol{a}(t_1,\vartheta_1), \boldsymbol{a}(t_2,\vartheta_2), \ldots, \boldsymbol{a}(t_{2n+1},\vartheta_{2n+1})].
\end{aligned}$$

The one-point exchange Remez algorithm for complex approximation is merely a straightforward mimic of Algorithm 1.

Algorithm 2

Step 0 Find an initial $\boldsymbol{t} \in [0,1]^{2n+1}$ and $\boldsymbol{\vartheta} \in [0, 2\pi]^{2n+1}$ such that $A(\boldsymbol{t},\boldsymbol{\vartheta})$ is nonsingular and $A^{-1}(\boldsymbol{t},\boldsymbol{\vartheta}) \cdot \boldsymbol{b} > \mathbf{o}$.

Step 1 Define $\boldsymbol{r}(\boldsymbol{t},\boldsymbol{\vartheta}) := A^{-1}(\boldsymbol{t},\boldsymbol{\vartheta}) \cdot \boldsymbol{b}$. From $(\boldsymbol{t},\boldsymbol{\vartheta})$ determine the unique $\boldsymbol{\sigma} := \begin{bmatrix} h \\ \boldsymbol{\lambda} \end{bmatrix}$ such that

$$\mathrm{Re}\Big(e^{-\iota\vartheta}[f(t_j) - p(\boldsymbol{\lambda}, t_j)]\Big) = h, \qquad j = 1, 2, \ldots, 2n+1.$$

Step 2 Determine t' such that $|f(t') - p(\boldsymbol{\lambda}, t')| = \|f - p(\boldsymbol{\lambda}, \cdot)\|$. Define the angle $\vartheta' := \mathrm{Arg}(f(t') - p(\boldsymbol{\lambda}, t'))$. If $\|f(t') - p(\boldsymbol{\lambda}, t')\| - h$ is small, h and $\boldsymbol{\lambda}$ are good approximations to the optimal solution and we terminate the algorithm. Otherwise, we move on.

Step 3 Exchange (t', ϑ') with one of the (t_j, ϑ_j)'s, $j = 1, 2, \ldots, 2n+1$ as done in the previous algorithm for the real case. Go back to Step 1.

4 Convergence of the Algorithm

The Remez algorithm, whether real or complex, generates a sequence of iterates $\boldsymbol{r}^{(k)}$, $(\boldsymbol{t}^{(k)}, \boldsymbol{\vartheta}^{(k)})$, and $\sigma^{(k)} = \begin{bmatrix} h^{(k)} \\ \boldsymbol{\lambda}^{(k)} \end{bmatrix}$. Questions of convergence are naturally centered on the quantity

$$\eta_k := \left\| f - p(\boldsymbol{\lambda}^{(k)}, \cdot) \right\| - h^{(k)}.$$

Does η_k converge to 0, and, if so, how fast? We answer these questions below in order.

4.1 Global convergence

In the discussion of the real Remez algorithm, [7] shows that $\eta_k \to 0$. The proof is a direct consequence of the fact that $\boldsymbol{r}^{(k)} \geq \Delta > 0$ for some uniform lower bound Δ for all the $\boldsymbol{r}^{(k)}$. Such a bound does not exist in general, however, for two (or higher) dimension real approximation or complex approximation. Nevertheless, similar to the situation in the simplex algorithm, it is reasonable to assume $\boldsymbol{r}^{(k)} > \mathbf{o}$ for all k.

Theorem 1 *If $\boldsymbol{r}^{(k)} > \mathbf{o}$ for $k = 1, 2, 3, \ldots$, then*

$$\liminf_{k \to \infty} \eta_k = 0.$$

Proof A complete proof can be found in [11]. ∎

In practice, $\liminf_{k \to \infty} \eta_k = 0$ means that the algorithm terminates in a finite number of steps for any positive stopping criterion.

4.2 Local convergence

For the real Remez algorithm, it is proved in [7] that the sequence $\{\eta_k\}$ is majorized by a sequence $\{\delta_k\}$, $\delta_k \geq \eta_k$ for all k, that converges to zero quadratically. The standard assumption ([1],[7], and [13]) for real approximation is that the optimal $f - p$ has $n + 1$ extrema (alternations) at which the second derivatives are nonzero. The assumption and convergence result are generalized to complex approximation in the next theorem.

Theorem 2 *Let the function $|f(t) - p(\boldsymbol{\lambda}^*, t)|$ have exactly $2n + 1$ extrema $t_j{}^*$, $1 \leq j \leq 2n + 1$, and let $\boldsymbol{r}^* := A^{-1}(\boldsymbol{t}^*, \boldsymbol{\vartheta}^*) \cdot \boldsymbol{b} > \mathbf{o}$, where $\vartheta_j{}^* := \operatorname{Arg}\big(f(t_j{}^*) - p(\boldsymbol{\lambda}^*, t_j{}^*)\big)$, $1 \leq j \leq 2n + 1$. Furthermore, let the second derivatives with respect to t of $|\ f(t) - p(\boldsymbol{\lambda}^*, t)\ |$ be nonzero at each of the $2n + 1$ extrema. Then there exists a sequence $\{\delta_k\}, \delta_k \geq \eta_k$ for all k, and two constants M and K such that*

$$\delta_{2n+1+k} \leq M \delta_k^2 \qquad \textit{for } k \geq K.$$

Proof The complete proof is found in [11]. ∎

Unfortunately, The proof in [11] is too long to be included here. That length, however, is mainly due to the fact that the algorithm is a one-point exchange instead of a multiple-exchange one. In the rest of this section, we first explain why quadratic convergence is obtainable. Then, we present a multiple exchange version of the Remez algorithm. Although this version works only when the iterate is close enough to the optimum, the proof of its quadratic convergence is sufficiently short to be presented here.

Consider the nonlinear system of equations

$$
\begin{array}{lrcll}
(4.1) & \mathrm{Re}\Big(e^{-\iota\vartheta_j}[f(t_j)-p(\lambda,t_j)]\Big)-h & = & 0 & j=1,2,\ldots,2n+1 \\
 & \frac{d}{dt}|f(t)-p(\lambda,t)|\,\Big|_{t_j} & = & 0 & j=1,2,\ldots,2n+1 \\
 & \mathrm{Im}\Big(e^{-\iota\vartheta_j}[f(t_j)-p(\lambda,t_j)]\Big) & = & 0 & j=1,2,\ldots,2n+1
\end{array}
$$

Denote the $3(2n+1)$ real parameters by groups of $2n+1$ thus $\boldsymbol{x} := (\boldsymbol{\sigma},\boldsymbol{t},\boldsymbol{\vartheta})$. Moreoever, denote the three groups of $2n+1$ lefthand sides by $\boldsymbol{g}_j(\boldsymbol{x}), j=1,2,3$. Finally, define $G(\boldsymbol{x}) := [\boldsymbol{g}_1^T(\boldsymbol{x}),\boldsymbol{g}_2^T(\boldsymbol{x}),\boldsymbol{g}_3^T(\boldsymbol{x})]$. Suppose the assumptions of Theorem 2 are satisfied; then it is not hard to show that $\boldsymbol{x}^* := (\boldsymbol{\sigma}^*,\boldsymbol{t}^*,\boldsymbol{\vartheta}^*)$ is an isolated zero of Equation 4.1. Moreover, we can prove the next theorem.

Theorem 3 *With the assumptions of Theorem 2, the Jacobian $J(\boldsymbol{x}^*)$ of G at $\boldsymbol{x}^*$ is invertible.*

Proof Partition the Jacobian into 3×3 blocks of dimension $2n+1$ each. Because of the structure of G, the derivatives with respect to $\boldsymbol{t}$ and $\boldsymbol{\vartheta}$ are all diagonal matrices. Moreover, because $\boldsymbol{t}^*$ and $\boldsymbol{\vartheta}^*$ are extrema of $\mathrm{Re}(e^{-\iota\vartheta}[f(t)-p(\lambda^*,t)])$ and $|f(t)-p(\lambda^*,t)|$,

$$
\begin{array}{lcl}
(\boldsymbol{g}_1)_{\boldsymbol{t}}(\boldsymbol{x}) & = & (\boldsymbol{g}_1)_{\boldsymbol{\vartheta}}(\boldsymbol{x}) = \mathbf{o} \quad \text{for all } \boldsymbol{x}, \\
(\boldsymbol{g}_2)_{\boldsymbol{t}}(\boldsymbol{x}^*) & = & \mathrm{diag}\Big(\frac{\partial^2}{\partial t^2}|f(t)-p(\lambda^*,t)|\,\Big|_{t_1^*},\ldots,\frac{\partial^2}{\partial t^2}|f(t)-p(\lambda^*,t)|\,\Big|_{t_{2n+1}^*}\Big), \\
(\boldsymbol{g}_3)_{\boldsymbol{\vartheta}}(\boldsymbol{x}^*) & = & \mathrm{diag}\Big(-h^*,-h^*,\ldots,-h^*\Big).
\end{array}
$$

Finally, $(\boldsymbol{g}_1)_{\boldsymbol{\sigma}}(\boldsymbol{x}^*) = -A^T(\boldsymbol{t}^*,\boldsymbol{\vartheta}^*)$. Clearly, the Jacobian at $\boldsymbol{x}$ is block lower triangular with the diagonal blocks invertible. ∎

It is easy to check that the Jacobian is Lipschitz continuous in $\boldsymbol{x}$ and thus Newton iteration would converge quadratically. This approach is not taken in real approximation, however, because of its high cost since the Newton's approach is of dimension $2(n+1)$ instead of $n+1$, the dimension of Remez. The situation is similar in complex approximation. Indeed, we can show that the following multiple exchange complex Remez algorithm, of dimension $2n+1$, converges as fast as Newton iteration does, which is of dimension $3(2n+1)$. The multiple exchange complex Remez algorithm is Algorithm 2 with Steps 2 and 3 replaced as follows:

Step 2′ Determine $t_1', t_2', \ldots, t_{2n+1}'$ near $t_1, t_2, \ldots, t_{2n+1}$, respectively, such that the first derivatives of $|f(t)-p(\lambda,t)|$ with respect to t vanish. Define ϑ_j' as the arguments of $f(t_j')-p(\lambda,t_j')$.

Step 3′ Replace $(\boldsymbol{t},\boldsymbol{\vartheta})$ by $(\boldsymbol{t}',\boldsymbol{\vartheta}')$.

This algorithm works when $(\boldsymbol{t},\boldsymbol{\vartheta})$ is close to the optimum because $A^{-1}(\boldsymbol{t}^*,\boldsymbol{\vartheta}^*)\cdot\boldsymbol{b} > \mathbf{o}$ implies $A^{-1}(\boldsymbol{t},\boldsymbol{\vartheta})\cdot\boldsymbol{b} > \mathbf{o}$ whenever $(\boldsymbol{t},\boldsymbol{\vartheta})$ is close to $(\boldsymbol{t}^*,\boldsymbol{\vartheta}^*)$.

Now, to show the equivalent convergence rate of the Newton and Remez iterations, we consider the updates $\boldsymbol{x}^{(\mathcal{N})}$ and $\boldsymbol{x}^{(\mathcal{R})}$ after one iteration of the Newton and the Remez algorithms, respectively, both starting at a common point $\boldsymbol{x}$ very close to the optimum $\boldsymbol{x}^*$, $\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \varepsilon$. We will show that $\|\boldsymbol{x}^{(\mathcal{N})} - \boldsymbol{x}^{(\mathcal{R})}\| < M\varepsilon^2$ for some constant M independent of ε.

First, the Newton update $\boldsymbol{x}^{(\mathcal{N})}$ is defined by

$$(4.2) \qquad J(\boldsymbol{x}) \cdot (\boldsymbol{x} - \boldsymbol{x}^{(\mathcal{N})}) = G(\boldsymbol{x}).$$

Next, consider $\boldsymbol{x}^{(\mathcal{R})}$. The vector $\boldsymbol{\sigma}^{(\mathcal{R})}$ is defined by

$$(g_1)_{\boldsymbol{\sigma}}(\boldsymbol{x}) \cdot (\boldsymbol{\sigma} - \boldsymbol{\sigma}^{(\mathcal{R})}) = g_1(\boldsymbol{x}).$$

Based on $\boldsymbol{\sigma}^{(\mathcal{R})}$, $\boldsymbol{t}^{(\mathcal{R})}$ and $\boldsymbol{\vartheta}^{(\mathcal{R})}$ are determined by

$$g_2(\boldsymbol{x}^{(\mathcal{R})}) = g_3(\boldsymbol{x}^{(\mathcal{R})}) = \mathbf{o}.$$

Thus,

$$\begin{bmatrix} g_2(\boldsymbol{x}) \\ g_3(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} g_2(\boldsymbol{x}) \\ g_3(\boldsymbol{x}) \end{bmatrix} - \begin{bmatrix} g_2(\boldsymbol{x}^{(\mathcal{R})}) \\ g_3(\boldsymbol{x}^{(\mathcal{R})}) \end{bmatrix} = \frac{\partial}{\partial \boldsymbol{x}} \begin{bmatrix} g_2(\boldsymbol{x}) \\ g_3(\boldsymbol{x}) \end{bmatrix} \cdot (\boldsymbol{x} - \boldsymbol{x}^{(\mathcal{R})}) + \boldsymbol{y},$$

where $\|\boldsymbol{y}\| < M_1\varepsilon^2$ for some M_1 indepedent of ε. Thus, $\boldsymbol{x}^{(\mathcal{R})}$ can be characterized as

$$(4.3) \qquad \left(J(\boldsymbol{x}) - \begin{bmatrix} \mathbf{o} & (g_1)_{\boldsymbol{t}}(\boldsymbol{x}) & (g_1)_{\boldsymbol{\vartheta}}(\boldsymbol{x}) \\ \mathbf{o} & \mathbf{o} & \mathbf{o} \\ \mathbf{o} & \mathbf{o} & \mathbf{o} \end{bmatrix} \right) \cdot \left(\boldsymbol{x} - \boldsymbol{x}^{(\mathcal{R})} \right) = G(\boldsymbol{x}) + \boldsymbol{y}.$$

Subtracting Equation 4.3 from Equation 4.2 gives

$$(4.4) \qquad J(\boldsymbol{x}) \cdot (\boldsymbol{x}^{(\mathcal{R})} - \boldsymbol{x}^{(\mathcal{N})}) = \boldsymbol{z}$$

where $\|\boldsymbol{z}\| < M_2\varepsilon^2$ for some M_2 independent of ε. This is because $\|(g_1)_{\boldsymbol{t}}(\boldsymbol{x})\|$, $\|(g_1)_{\boldsymbol{\vartheta}}(\boldsymbol{x})\|$, and $\|\boldsymbol{x} - \boldsymbol{x}^{(\mathcal{R})}\| < M_3\varepsilon$ for some M_3 independent of ε. Finally, $J(\boldsymbol{x}^*)$ is invertible, and thus $\left\|J^{-1}(\boldsymbol{x})\right\|$ is uniformly bounded near $\boldsymbol{x}^*$ and the proof is complete.

Numerical examples illustrating the convergence behavior can be found in [12] and [11].

5 An Extension of the Complex Remez Algorithm

In our experience ([11] and [12]), whenever the assumptions in Theorem 2 are not satisfied, the number of extrema is insufficient. Fortunately, in that case we can extend the algorithm slightly to restore quadratic convergence. Details of this extension are presented in [11] and [12]; the idea is roughly as follows. If the number of extrema is d fewer than the ideal number, one can define a function $\boldsymbol{u} : \mathbf{R}^d \rightarrow \mathbf{R}^d$ whose zero corresponds to the solution of the original approximation problem. Moreover, to evaluate the function and its Jacobian, we need only to apply the complex Remez algorithm to a problem *that satisfies all the conditions needed for fast convergence*. Consequently, Newton iteration applied on $\boldsymbol{u}$ is efficient. The quadratic convergence of this extended algorithm is fully analyzed in [11].

6 Conclusion

We have shown that linear complex Chebyshev approximation can be naturally viewed as a special case of two-dimensional real approximation for which the Remez exchange algorithm converges quadratically. We believe such a view can help our understanding in other aspects of complex approximation such as Chebyshev approximation by rational functions.

7 Acknowledgment

This work was supported in part by the Strategic Defense Initiative Organization, Office of the Secretary of Defense, under WPD B411.

References

[1] E. W. Cheney. *Introduction to Approximation Theory.* Chelsea, New York, 1986.

[2] H. C. Elman and R. L. Streit. *Polynomial Iteration for Nonsymmetric Indefinite Linear Systems.* Research report YALEU/DCS/RR-380, Department of Computer Science, Yale University, New Haven, Conn., March 1985.

[3] B. Francis, J. W. Helton, and G. Zames. H^∞-optimal feedback controllers for linear multivariable systems. *IEEE Trans. Autom. Control*, 29:888–900, October 1984.

[4] K. Glashoff and K. Roleff. A new method for Chebyshev approximation of complex-valued functions. *Mathematics of Computation*, 36(153):233–239, January 1981.

[5] M. Hartmann and G. Opfer. Uniform approximation as a numerical tool for constructing conformal maps. *Journal of Computational and Applied Mathematics*, 14:193–206, 1986.

[6] J. W. Helton. Worst case analysis in the frequency domain: An H^∞ approach for control. *IEEE Trans. Auto. Control*, 30:1192–1201, December 1985.

[7] M. J. D. Powell. *Approximation Theory and Methods.* Cambridge University Press, Cambridge, 1981.

[8] L. Reichel. On polynomial approximation in the complex plane with application to conformal mapping. *Mathematics of Computation*, 44(170):425–433, April 1985.

[9] T. J. Rivlin and H. S. Shapiro. A unified approach to certain problems of approximation and minimization. *Journal of Soc. Indust. Appl. Math.*, 9(4):670–699, December 1961.

[10] R. L. Streit and A. H. Nuttall. A general Chebyshev complex function approximation procedure and an application to beamforming. *Journal of Acoustical Society of America*, 72:181–190, July 1982.

[11] P. T. P. Tang. *Chebyshev Approximation on the Complex Plane.* PhD thesis, Department of Mathematics, University of California at Berkeley, May 1987.

[12] P. T. P. Tang. A fast algorithm for linear complex Chebyshev approximations. *Mathematics of Computation*, 51(184):721–739, October 1988.

[13] L. Veidinger. On the numerical determination of the best approximation in the Chebyshev sense. *Numerische Mathematik*, 2:99–105, 1960.